

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M2612 – Elektrotechnika a informatika

Studijní obor: 3906T001 – Mechatronika

Návrh inteligentního HW rozhraní pro řízení krokových motorů v úlohách automatizovaného měření

**Design of intelligent HW interface for stepper motor control
for using in automated measurement**

DIPLOMOVÁ PRÁCE

Autor:

Martin Šrajer

Vedoucí práce:

Ing. Tomáš Mikolanda

Konzultant:

Ing. Jan Václavík

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra elektrotechniky

Akademický rok: 2006/2007

ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení: Martin Šrajer

studijní program: M 2612 – Elektrotechnika a informatika

obor: 3906T001 - Mechatronika

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto diplomovou práci:

Název tématu: Návrh inteligentního HW rozhraní pro řízení krokových motorů v úlohách automatizovaného měření

Zásady pro vypracování:

1. Zanalyzujte a proveďte rozbor existujících principů řízení krokových motorů.
2. Proveďte simulaci vlivu napětí na velikost a průběh proudu vinutím krokového motoru a jeho moment. Porovnejte vlivy napětí na moment motorku v závislosti na jeho úhlové rychlosti s ohledem na použití v aplikacích lineárních posuvů, souřadnicové systémy.
3. Na základě analýzy a simulací navrhnete modul výkonového řízení krokového motoru s možností programovatelného nastavení úhlového zrychlení, rychlosti a polohy se zaměřením na použití v úlohách automatizovaného měření.
4. Zkonstruujte zařízení pro simultánní a nezávislé řízení třech krokových motorů. Navrhnete a zkonstruujete inteligentní rozhraní pro ovládání z PC s možností budoucího SW rozšíření. Navrhnete, implementujete a zdokumentujete vhodný komunikační protokol pro vzdálené řízení s možností autonomního chodu. Kladte po SW-firmware strážce důraz na použití motorů s lineárními pohony ve funkci 2D polohovacího systému, tzn. pohyb po přímce, pohyb po křivce.
5. Vytvořte jednoduchý SW pro PC určený k ověření základních parametrů realizovaného systému a otestování jeho funkčnosti.
6. Vytvořte technickou dokumentaci k realizovanému systému.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 40 až 50 stran

Seznam odborné literatury:

[1] Albert C. Leenhouts: Step Motor System Design Handbook, Magna Physics Publishing, ISBN 1-881855-08-2

[2] Novák Petr: Mobilní roboty - pohony, senzory, řízení. BEN – technická literatura, Praha 2005, ISBN 80-7300-141-1

[3] Krejčířík, A.: Napájecí zdroje I, II, III. BEN - technická literatura, Praha 1998, ISBN 80-86056-56-2

[4] Benda V.: Výkonové polovodičové součástky a integrované struktury; 1. vydání. Praha: ČVUT. 1994, ISBN 80-01-01140-2

Vedoucí diplomové práce: Ing. Tomáš Mikolanda

Konzultant: Ing. Jan Václavík

Zadání diplomové práce: 23.10.2006

Termín odevzdání diplomové práce: **18. 5. 2007**

L.S.

.....
Vedoucí katedry

.....
Děkan

V Liberci dne 23.10.2006

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Na tomto místě bych rád poděkoval vedoucímu diplomové práce Ing. Tomáši Mikolandovi a konzultantu Ing. Janu Václavíkovi za jejich odborné rady, konzultace a pomoc při tvorbě této práce. Dále bych rád poděkoval své přítelkyni Áje za to, že za mnou stála i v nelehkých životních situacích a nemalou měrou přispěla k jejich zvládnutí.

Abstrakt

Cílem této diplomové práce byl návrh a realizace řízení tříosého polohovacího systému. Jako pohon byly použity krokové motory, proto je úvodní část práce věnována jejich popisu a simulaci chování. Cílem simulací byl výběr nejvhodnějšího způsobu řízení motorů. První praktická část práce byla zaměřena na realizaci řídicí jednotky krokových motorů a tvorbu jeho software. Řídicí jednotka byla úspěšně navržena a je založena na mikrokontroléru ATMEL ATmega16. Software jednotky mimo jiné umožňuje lineární řízení rychlosti otáčení krokového motoru.

Další úlohou byla realizace řídicí jednotky polohovacího systému. Vytvořená jednotka, založená na mikrokontroléru ATmega128, přijímá požadavky polohování od uživatele, na jejich základě generuje parametry pohybu a zasílá je jednotlivým jednotkám motorů. Též ovládá LCD displej a obsluhuje vstup z numerické klávesnice.

Závěrečnou úlohou bylo oživení celého systému a demonstrace jeho ovládání pomocí vlastní aplikace pod operačním systémem Windows.

Klíčová slova: krokový motor, polohovací systém, mikrokontrolér ATmega.

Abstract

The aim of this diploma thesis was the design of control system for 3D positioning device. The stepper motors are used as a drive of the device therefore the first part of this work concerns about their description and about their behavior simulations. The goal of the simulations is to choose the best suitable way of stepper motor control. The first practical task was the realization of universal electronic control unit for stepper motors. The successfully designed unit is based on the ATMEL ATmega16 microcontroller. SW of the unit provides linear speed control of the motor.

The next task was the positioning system control unit realization and its SW development. The created unit is based on ATmega128 microcontroller and its primary goal is user commands receiving, their recognition and movement parameters generation. The demands of movement are then sent to motor units.

The final task was the PC (WIN) application development which provides graphical user interface. User is then able to have control over the whole system through this PC application.

Keywords: stepper motor, positioning device, ATmega microcontroller.

Obsah

ABSTRAKT	5
SEZNAM OBRÁZKŮ	8
SEZNAM POUŽITÝCH SYMBOLŮ	9
ÚVOD	11
1 POPIS KROKOVÝCH MOTORŮ A JEJICH ŘÍZENÍ.....	12
1.1 KROKOVÉ MOTORY	12
1.2 KONSTRUKCE KROKOVÝCH MOTORŮ.....	13
1.2.1 Krokové motory s proměnnou reluktancí.....	13
1.2.2 Krokové motory s permanentními magnety.....	13
1.2.3 Krokové motory hybridní	14
1.3 ZPŮSOBY ŘÍZENÍ KROKOVÝCH MOTORŮ	14
1.3.1 Řízení jednofázové, dvoufázové.....	15
1.3.2 Unipolární řízení.....	15
1.3.3 Bipolární řízení.....	16
1.4 MOŽNOSTI ZAPOJENÍ VINUTÍ MOTORU	16
1.5 ŘÍZENÍ S PLNÝM A POLOVIČNÍM KROKEM	18
1.6 MIKROKROKOVÁNÍ.....	19
2 SIMULACE V MATLAB/SIMULINK	21
2.1 NÁHRADNÍ OBVOD FÁZE MOTORU	21
2.1.1 Sériový L-R obvod.....	22
2.1.2 Závislost velikosti a průběhu proudu na rychlosti krokování	25
2.1.3 Závislost momentu motoru na proudu procházejícím vinutím	26
2.2 MOŽNOSTI ŘÍZENÍ PROUDU VINUTÍM	26
2.2.1 Řízení proudu s pevnou dobou vypnutí spodního tranzistoru	27
2.2.2 Řízení proudu vinutím pomocí pulsně-šířkové modulace (PWM)	28
2.3 SIMULAČNÍ MODEL	30
2.4 VÝSLEDKY SIMULACÍ	32
2.5 ZÁVĚRY ZE SIMULACÍ A TEORETICKÝCH POZNATKŮ	35
2.5.1 Použité krokové motory a volba typu řízení	35
2.5.2 Problémy s rezonancí motoru	36
3 REALIZACE ŘÍDÍCÍ JEDNOTKY KROKOVÝCH MOTORŮ.....	40
3.1 LOGICKÁ ČÁST JEDNOTKY	40
3.1.1 Mikrokontrolér.....	40
3.1.2 Komunikace mikrokontroléru s okolím.....	41
3.1.3 Blokové schéma logické části jednotky	44
3.2 VÝKONOVÁ ČÁST JEDNOTKY	48
3.2.1 Výběr spínacích tranzistorů	48
3.2.2 Řízení unipolárních tranzistorů	49
3.2.3 Napájení řídicí jednotky.....	50
3.3 SOFTWARE ŘÍDÍCÍ JEDNOTKY	51
3.3.1 Struktura projektu	51
3.3.2 Lineární řízení rychlosti otáčení krokového motoru	52
3.3.3 Algoritmus výpočtu časové konstanty pro lineární řízení rychlosti	53
3.3.4 Generování rychlostního profilu.....	54
3.3.5 Vývojové diagramy programového vybavení řídicí jednotky	56
4 ŘÍDÍCÍ JEDNOTKA POLOHOVACÍHO SYSTÉMU	58
4.1 TOPOLOGIE CELÉHO ŘÍDÍCÍHO SYSTÉMU.....	59
4.2 NÁVRH ELEKTRONIKY JEDNOTKY	60
4.2.1 Výběr mikrokontroléru.....	60
4.2.2 Blokové schéma řídicí jednotky.....	61
4.3 SOFTWARE ŘÍDÍCÍ JEDNOTKY	66

4.3.1	Struktura projektu	66
4.3.2	Obsah jednotlivých zdrojových souborů	66
4.4	VÝVOJOVÉ DIAGRAMY SW	68
4.5	PŘÍKAZY PŘEDÁVANÉ UŽIVATELEM	70
4.5.1	SCPI standard	70
4.5.2	Soubor realizovaných příkazů a jejich popis	71
4.6	PŘEDÁVÁNÍ ZPRÁV V RÁMCI CELÉHO SYSTÉMU	73
4.7	DESKY PLOŠNÝCH SPOJŮ	75
5	OVLÁDACÍ SW PRO PC	76
5.1	TVORBA A POPIS APLIKACE	76
5.1.1	Zdrojový kód	76
5.1.2	Grafické rozhraní	77
6	ZÁVĚR	79
	LITERATURA	81

Přílohy

PŘÍLOHA A	83
ŘÍDÍCÍ JEDNOTKA KROKOVÉHO MOTORU – LOGICKÁ ČÁST	83
ŘÍDÍCÍ JEDNOTKA KROKOVÉHO MOTORU – VÝKONOVÁ ČÁST	86
PŘÍLOHA B	88
ŘÍDÍCÍ JEDNOTKA POLOHOVACÍHO SYSTÉMU	88
PŘÍLOHA C	92
FOTODOKUMENTACE	92
PŘÍLOHA D	97
STRUKTURA A OBSAH PŘILOŽENÉHO CD	97

Seznam obrázků

Obrázek 1.1	Zapojení pro unipolární řízení.	15
Obrázek 1.2	Časová souslednost spínacích impulsů	16
Obrázek 1.3	Krokový motor - vyvedení vinutí	16
Obrázek 1.4	Sériové zapojení cívek vinutí.....	17
Obrázek 1.5	Časová souslednost spínacích impulsů (bipolární, dvoufázové řízení) ..	17
Obrázek 1.6	Paralelní zapojení cívek vinutí.....	18
Obrázek 1.7	Proud jednou fází motoru při mikrokrokování	19
Obrázek 2.1	Sériový L-R obvod.....	21
Obrázek 2.2	Příklad přechodové charakteristiky	23
Obrázek 2.3	Závislost doby potřebné k dosažení jmen. proudu na nap. napětí	24
Obrázek 2.4	Závislost velikosti a průběhu proudu na rychlosti krokování.....	25
Obrázek 2.5	Principiální zapojení pro řízení proudu s pevnou dobou vypnutí.....	27
Obrázek 2.6	Popis časových úseků PWM signálu	28
Obrázek 2.7	Příklad generování PWM signálu	29
Obrázek 2.8	Blokové schéma simulačního modelu	30
Obrázek 2.9	Proudy vinutím a výsledný točivý moment motoru.....	32
Obrázek 2.10	Průběh proudu vinutím při mikrokrokování	33
Obrázek 2.11	Průběh proudu vinutím a moment motoru při mikrokrokování.....	34
Obrázek 2.12	Průběh statického momentu.....	36
Obrázek 2.13	Průběh proudu při rezonanci	37
Obrázek 2.14	Změny v průběhu proudu při zvyšování f_k	38
Obrázek 2.15	Příklad momentové charakteristiky dvoufáz. krokového motoru.....	38
Obrázek 3.1	Návrh koncepce řídicí jednotky	40
Obrázek 3.2	Propojení jednotek na SPI sběrnici (single master/single slave)	42
Obrázek 3.3	Princip SPI komunikace.....	42
Obrázek 3.4	Zapojení SPI sběrnice při multi-slave konfiguraci	43
Obrázek 3.5	Blokové schéma logické části jednotky	44
Obrázek 3.6	Zapojení elektronického přepínače	45
Obrázek 3.7	Zapojení elektronické pojistky.....	47
Obrázek 3.8	Blokové schéma výkonové části řídicí jednotky	48
Obrázek 3.9	Nabíjení <i>bootstrap</i> kondenzátoru během sepnutí spodního tranzistoru..	50
Obrázek 3.10	Struktura souborů v projektu	51
Obrázek 3.11	Příklad průběhu kinematických veličin pohonu	52
Obrázek 3.12	Zpoždění mezi pulsy	53
Obrázek 3.13	Příklad rychlostního profilu, kdy není dosaženo požadované rychlosti .	55
Obrázek 3.14	Vývojový diagram obsluhy přerušení časovače T1	57
Obrázek 4.1	Blokové schéma řídicí jednotky polohovacího systému.....	61
Obrázek 4.2	Ilustrace funkce optické závory	63
Obrázek 4.3	Ilustrace zapojení maticové klávesnice.....	64
Obrázek 4.4	Rozdělení projektu do souborů	66
Obrázek 5.1	Grafické rozhraní aplikace.....	77

Seznam použitých symbolů

Symbol	Jednotka	Popis
α	rad	velikost kroku motoru
C_s		konstanta stejnosměrného stroje
ε	$\frac{rad}{s^2}$	úhlové zrychlení
f_k	Hz	frekvence krokování
f	Hz	frekvence signálu
f_T	Hz	frekvence časovače
h	$N.m$	statický moment krokového motoru
i	A	okamžitá hodnota proudu
I	A	proud
I_{jm}	A	jmenovitý proud motorem
I_D	A	proud drain-source MOSFET tranzistoru
J	kgm^2	moment setrvačnosti
J_r	kgm^2	moment setrvačnosti rotoru
K		statické zesílení dynamické soustavy
L	H	indukčnost
M_h	$N.m$	hnací moment motoru
M_z	$N.m$	zatěžovací moment
R	Ω	elektrický odpor, rezistance
$R_{DS(on)}$	Ω	odpor MOSFET tranzistoru v sepnutém stavu
t	s	čas
t_s	s	doba ustálení přechodového jevu
T	s	perioda signálu

<u>Symbol</u>	<u>Jednotka</u>	<u>Popis</u>
τ	s	časová konstanta
U	V	elektrické napětí
U_{DS}	V	napětí drain-source MOSFET tranzistoru
Φ	Wb	magnetický indukční tok
Θ	rad	poloha hřídele motoru
ω	$\frac{rad}{s}$	úhlová rychlost

Úvod

Na Katedře elektrotechniky TUL bylo potřeba vyřešit problém automatizovaného měření stacionárních magnetických polí prostorových (transformátory) a plošných objektů (magnety pro průmyslovou praxi). Záměrem bylo dosáhnout vyšší kvality, přesnosti a reprodukovatelnosti naměřených dat, které nelze ručním měřením zaručit. Za účelem realizace automatizovaného měřicího pracoviště byl již v minulosti zakoupen elektromechanický polohovací systém a krokové motory s řídicími jednotkami. Ukázalo se však, že řídicí jednotky krokových motorů nevyhovují všem požadavkům na ně kladeným. Hlavním nedostatkem je jejich „nulová inteligence“ – veškeré výpočetní algoritmy pro řízení polohy, rychlosti a zrychlení musí být realizovány pomocí nadřazeného PC s multifunkční kartou. Problémem se též ukázala být absence jakékoli ochrany proti přetížení.

Výše uvedené problémy byly motivem pro tuto diplomovou práci. Jejím cílem bylo navrhnout a realizovat elektroniku a řízení polohovacího systému zmíněného měřicího pracoviště. Základním úkolem bylo zkonstruovat řídicí jednotky krokových motorů, které by svými parametry předčily v současnosti používané komerčně vyráběné jednotky. Při návrhu mělo být využito simulací v prostředí MATLAB/SIMULINK.

Dalším úkolem byla konstrukce řídicí jednotky celého polohovacího systému, jež má být nadřazena jednotkám krokových motorů. Povelů od uživatele mají být zadávány ze vzdáleného PC (dle standartu SCPI), ale též má být umožněn autonomní režim ovládání. V tomto režimu má být uživateli umožněno vkládat povelů pomocí numerické klávesnice a k vizualizaci má být použit LCD displej.

Závěrečnou úlohou bylo naprogramovat aplikaci pro PC pod Windows, která uživateli umožní pohodlně zadávat požadavky polohování a bude demonstrovat možnosti celého realizovaného systému.

V některých částech této práce je k popisu využito anglických výrazů, protože tyto výrazy často nemají český ekvivalent nebo je jeho význam nepřesný.

1 Popis krokových motorů a jejich řízení

V této kapitole se zaměřím na popis krokových motorů, odlišnosti mezi různými konstrukčními typy a zejména na způsoby jejich řízení. K problematice pohonů s krokovými motory lze nalézt dostatek publikací např. [1] a [2]. Těžiště mé práce má spočívat zejména v praktické realizaci a proto je teorii krokových motorů věnován poměrně malý prostor.

1.1 Krokové motory

Z principiálního hlediska je krokový motor převodníkem elektrické energie na mechanickou. Krokové motory jsou používány jako výkonové prvky v elektrických pohonech pro nastavování polohy a rychlosti. Jejich oblast použití je rozsáhlá, lze je nalézt v textilním průmyslu, v periferiích počítačů a kancelářských zařízeních, ale také jako součást přesných polohovacích robotů. Dalšími aplikacemi jsou řízení otáček s realizací přímého pohonu včetně strojů, pohon různých čerpadel apod.

Jejich velkou výhodou je, že mohou pracovat bez zpětné vazby v otevřené smyčce, tedy není bezpodmínečně nutné odměřování polohy rotoru. To je zásadní výhoda zejména v jednodušších aplikacích, kde snímač polohy představuje značnou investici.

Pracují s minimální nutností údržby po celou dobu své životnosti. Charakteristickou vlastností motoru je otáčení hřídele po krocích. Otáčky hřídele jsou tedy nespojité. Jedna otáčka je složena z pevně definovaného počtu kroků, který odpovídá konstrukci a způsobu řízení. Počet stabilních poloh rotoru je tedy dán počtem kroků motoru na jednu otáčku.

V zásadě si můžeme krokový motor představit jako synchronní stroj v němž místo točivého pole, které je např. u běžného synchronního motoru generováno třífázovým napájecím napětím statorového vinutí, je generováno "pulzující" magnetické pole postupným napájením jednotlivých pólových dvojic stejnosměrným proudem.

Další poměrně specifickou vlastností krokových motorů je maximální točivý moment již od nulových otáček, což je v mnoha aplikacích též velkou předností. Nevýhodou je naopak poměrně komplikovaná konstrukce a v porovnání se servopohony nízké maximální výkony.

1.2 Konstrukce krokových motorů

Existují tři základní varianty konstrukce krokových motorů a sice motory s *permanentními magnety* a motory s *proměnnou reluktancí*. Třetí variantou je kombinace obou předchozích, tyto motory jsou pak nazývány *hybridní*.

1.2.1 Krokové motory s proměnnou reluktancí

Stator i rotor tohoto typu krokového motoru jsou složeny z plechů. Neobsahují permanentní magnety. Mohou být tří i vícefázové. Činnost krokového motoru je založená na změnách reluktance magnetického obvodu stroje při otáčení rotoru. Při zanedbání reluktance železa vůči vzduchu, rozptylu a zakřivení magnetických indukčních čar ve vzduchové mezeře je reluktance obvodu fáze statoru lineární funkcí překrytí zubů statoru a rotoru. Indukčnost cívky fáze je tedy největší, jsou-li osy zubů rotoru a statoru proti sobě. Minimální reluktance je v poloze posunuté o polovinu rozteče. Při odpojení statoru od napájení nevykazují žádný přídržný moment. Základní krok obvykle závisí na počtu zubů rotoru.

Z důvodu jejich mohutnosti, hmotnosti a velké vlastní spotřeby se již příliš nepoužívají.

1.2.2 Krokové motory s permanentními magnety

Jejich rotor je tvořen permanentním magnetem, někdy bývají nazývány též motory s aktivním rotorem. Podle uspořádání pólů magnetu rozlišujeme dvě skupiny těchto motorů: s radiálně polarizovaným nebo s axiálně polarizovaným permanentním magnetem. Charakteristickým znakem je rozdílný počet pólů na statoru i rotoru, přičemž rotorové póly jsou tvořeny permanentními magnety.

Vyznačují se nízkou cenou, ale také pouze malým točivým momentem a malým počtem kroků na otáčku. Používají se zejména v periferiích počítačů (tiskárny).

1.2.3 Krokové motory hybridní

Rotor motoru tvoří axiálně uspořádané permanentní magnety a dvě části z magneticky měkkého železa, přecházející na přesazené pólové nástavce. Na statoru je v drážkách navinuto vinutí. V případě, že do vinutí není přiváděn proud, rotor zaujímá takovou pozici, v níž je nejmenší reluktance. Moment, který udržuje rotor v této pozici se nazývá *zbytkový moment*. Tyto motory mají malý základní krok, větší točivý moment a větší přídržný moment než oba předchozí typy. Většina těchto motorů je dvoufázových, méně často se používají pětifázové.

Hybridní motory jsou v současnosti nejpoužívanějším typem krokových motorů a jejich vývoj a zdokonalování i nadále probíhá.

1.3 Způsoby řízení krokových motorů

Způsoby řízení krok. motorů lze rozdělit dle různých kritérií. Jednotlivé vrstvy tohoto rozdělení jsou vzájemně kombinovány a dohromady tvoří velké množství možných způsobů řízení. Z důvodu stanoveného rozsahu práce není možné popsat veškeré možnosti a metody řízení, budu se tedy věnovat jen nejvýznamnějším z nich.

Základní dělení typů řízení:

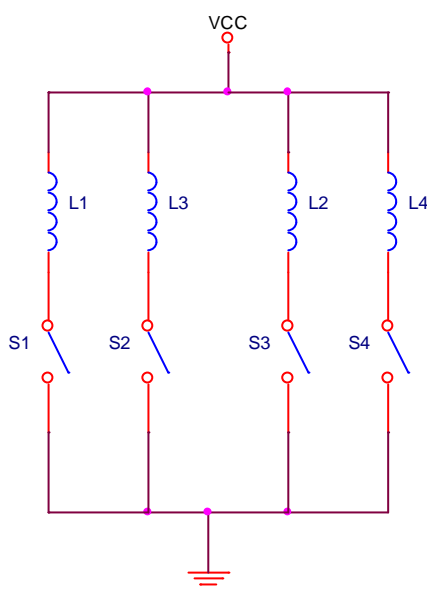
- Dle směru průchodu proudu vinutím:
 - Unipolární
 - Bipolární
- Dle počtu současně buzených fází:
 - Jednofázové
 - Dvoufázové
- Dle krokování:
 - Celo-krokování (Full stepping)
 - Polo-krokování (Half stepping)
 - Mikrokrokování

1.3.1 Řízení jednofázové, dvoufázové

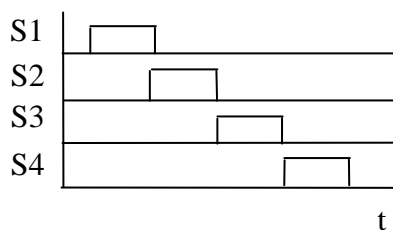
Při jednofázovém řízení je v jeden okamžik buzena vždy pouze jedna fáze vinutí motoru. Magnetické pole je tedy vytvářeno jen jednou fází. Při dvoufázovém způsobu řízení jsou buzeny vždy dvě fáze motoru. Je tedy zřejmé, že při dvoufázovém buzení můžeme dosáhnout vyššího točivého momentu, avšak spotřeba proudu bude dvojnásobná.

1.3.2 Unipolární řízení

Při tomto způsobu řízení prochází proud cívkou příslušné fáze motoru vždy jedním směrem a ta vytváří magnetické pole vždy stejné orientace. Tím, že je nabuzena vždy jen jedna cívka příslušné fáze je redukována indukčnost motoru. To se projeví pozitivně při vyšších rychlostech krokování. Výhodou tohoto způsobu řízení je menší spotřeba proudu oproti bipolárnímu řízení, vyšší maximální dosažitelná rychlost a jednodušší řízení avšak podstatnou nevýhodou je zhruba 40% ztráta točivého momentu.



Obrázek 1.1 Zapojení pro unipolární řízení. Spínače S1-S4 jsou připojeny k cívkám vinutí. Indukčnosti L1-L4, představující cívky 2-fázového krokového motoru (2 cívky pro každou fázi).



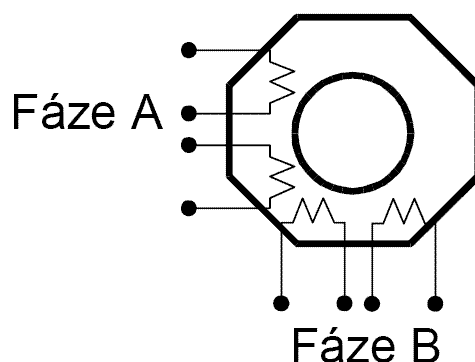
Obrázek 1.2 Časová souslednost spínacích impulsů ilustruje princip tohoto řízení: Postupným spínáním spínačů S1-S4 dojde k vytvoření pulsujícího magnetického pole a k roztočení rotoru.

1.3.3 Bipolární řízení

Při bipolárním řízení prochází proud vždy dvěma protilehlými cívkami. Ty jsou zapojené tak, aby jimi vytvořené magnetické pole bylo opačně orientované. K tomuto způsobu řízení je zapotřebí jednoho H-můstku se spínacími prvky (nejčastěji tranzistory) pro každou fázi motoru. Řízení je složitější a náklady na jeho realizaci vyšší než v případě unipolárního řízení z důvodu dvojnásobného počtu spínacích prvků. Avšak většina výrobců udává jmenovité hodnoty u svých motorů právě při uvažovaném bipolárním řízení. Pokud tedy chceme daný motor maximálně využít, zejména co do hodnoty točivého momentu, je nutné realizovat tento typ řízení.

1.4 Možnosti zapojení vinutí motoru

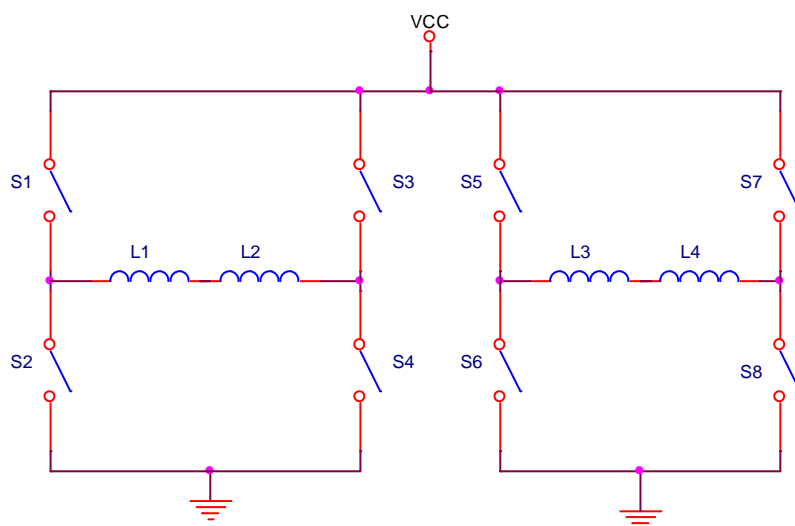
Na následujícím obrázku je schematicky znázorněn hybridní dvoufázový motor. Má-li motor takto vyvedena vinutí je nazýván *bifilárním* a existují při bipolárním řízení dvě odlišné varianty zapojení s různými vlastnostmi.



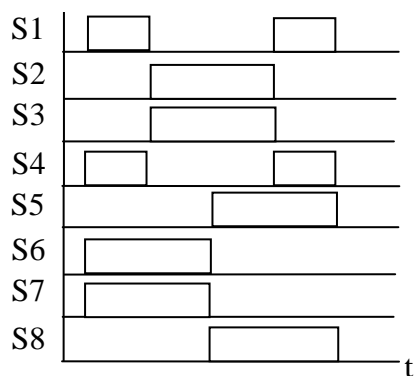
Obrázek 1.3 Krokový motor – vyvedení vinutí dle [3]

Sériové zapojení cívek vinutí

Cívky L1, L2 představující vinutí fáze A jsou spojeny do série, stejně jako L3, L4 ilustrující fázi B. Při tomto zapojení a bipolárním řízení dosáhneme v nízkých a středních otáčkách o 40% většího točivého momentu než při unipolárním řízení. Avšak indukčnost motoru je 4x vyšší než při zapojení pro unipolární řízení. To způsobí výrazný pokles točivého momentu při vyšších rychlostech.



Obrázek 1.4 Sériové zapojení cívek vinutí



Obrázek 1.5 Časová souslednost spínacích impulsů (bipolární, dvoufázové řízení)

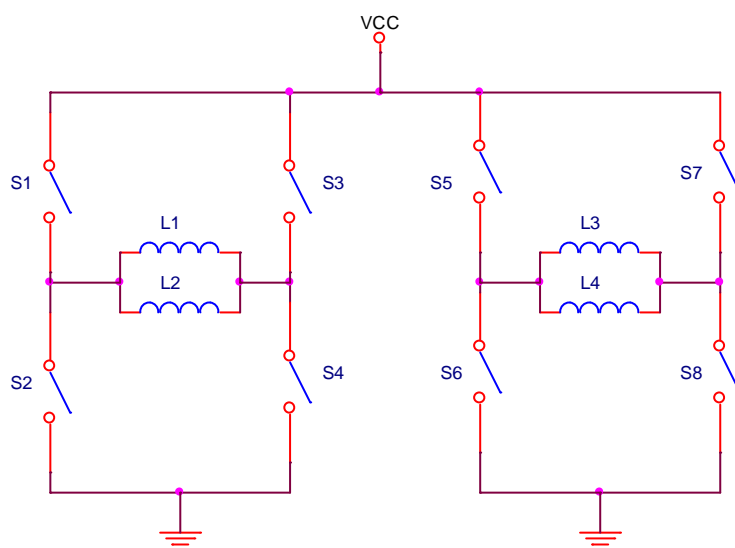
Z diagramu je patrné, že proud poteče v každý okamžik oběma fázemi a vzhledem k příslušnému vinutí se bude měnit směr jeho průchodu viz Tabulka 1.1

Tabulka 1.1 Znázornění směru průchodu proudu oběma fázemi motoru při spínání dle Obrázek 1.5.

	Krok	Fáze A	Fáze B
↓	1	+	-
	2	-	-
	3	-	+
	4	+	+
	1	+	-

Paralelní zapojení cívek vinutí

Cívky L1, L2 fáze A jsou zapojeny paralelně, taktéž L3, L4 fáze B. Stejně jako při sériovém zapojení cívek získáme o 40% vyšší točivý moment proti unipolárnímu řízení a to navíc v širokém rozmezí otáček. Limitujícím faktorem pro využití tohoto zapojení však může být dvojnásobná spotřeba proudu, tedy vyšší nároky na napájecí zdroj a také větší teplotní zatížení motoru.



Obrázek 1.6 Paralelní zapojení cívek vinutí

1.5 Řízení s plným a polovičním krokem

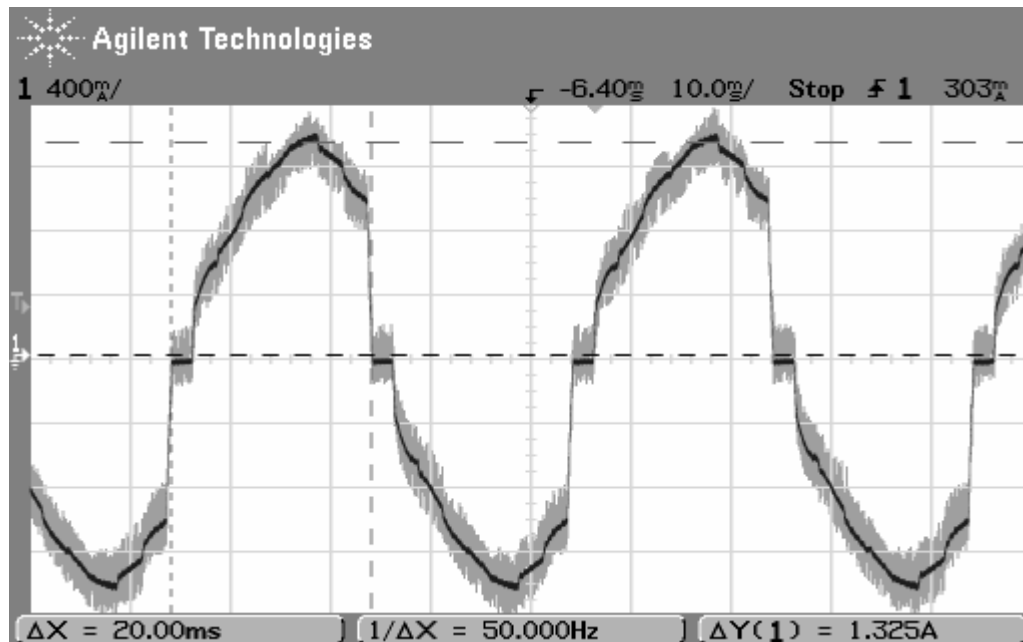
Dle časové souslednosti spínání proudu do jednotlivých cívek vinutí (jejich buzení) rozlišujeme dva základní typy krokování.

Při celokrokování má rotor tolik stabilních poloh v rámci jedné otáčky, kolik zubů má stator daného motoru. Tedy počet poloh rotoru je dán konstrukcí motoru.

Řízení motoru s polovičním krokem (half stepping) je založeno na kombinaci jednofázového a dvoufázového řízení, kdy proud prochází nejprve jednou cívkou a v dalším kroku dvěma sousedními cívkami. Tím, že jsou buzeny dvě sousední cívky, dojde k vytvoření magnetického pole, jehož působením je přitažen rotor do mezipolohy mezi dvěma základními polohami. Dochází tedy ke zmenšení základního kroku na polovinu a lze tak dosáhnout dvojnásobného rozlišení. Ovšem nevýhodou je nerovnováha momentu motoru (z důvodu střídání buzení jedné a dvou cívek), což se může v dynamicky náročnějších aplikacích projevit ztrátou kroku.

1.6 Mikrokrokování

Mikrokrokování je poměrně nový způsob řízení krokových motorů založený na myšlence plynulé změny proudu protékajícího vinutím motoru. Pokud v jedné fázi motoru plynule zvyšujeme proud a zároveň v sousední fázi proud snižujeme, bude se výsledný vektor magnetické indukce plynule pohybovat mezi oběma krajními polohami jednoho mechanického kroku (ten je dán konstrukcí motoru). U dvoufázového motoru má tedy průběh proudu jednou fází charakter sinusoidy, zatímco ve druhé fázi charakter kosinusoidy.



Obrázek 1.7 Proud jednou fází motoru při mikrokrokování

Důvodem, proč se mikrokrokování používá, je zejména odstranění problémů s rezonancí, protože plynulým nárůstem proudu vinutím dojde k omezení velikosti přechodových jevů.

Další předností je dosažení výrazně „hladšího“ pohybu při nízkých rychlostech (snaha o eliminaci nespojitého pohybu rotoru). Znatelně poklesne hluk vznikající při pohybu rotoru, který je způsobován velkými změnami proudu. Nevýhodou je složitější a náročnější řízení.

Mnoho komerčně vyráběných integrovaných obvodů pro řízení krokových motorů mikrokrokování umožňuje, tyto obvody jsou již finančně dostupné i pro jednodušší a lacinější aplikace. V těchto obvodech jsou často integrovány i spínací tranzistory a k jejich výstupu lze přímo připojit motor.

Po úvaze jsem se však rozhodl řešení s integrovaným obvodem nepoužít, protože tento způsob má též mnoho nevýhod, přičemž nejvýznamnější z nich je ztrátový výkon těchto obvodů. Výkonové tranzistory v takových integrovaných obvodech totiž zdaleka nedosahují parametrů diskretních tranzistorů a zejména při vyšším proudovém zatížení by se obvod velice zahříval.

2 Simulace v MATLAB/SIMULINK

Počítačové simulace jsou prostředkem získávání informací o chování reálných systémů. Požadavkem je, aby simulační model co možná nejvěrněji vystihoval realitu avšak při zachování jeho přijatelné složitosti. Nalezení rozumného kompromisu může být někdy poměrně obtížné.

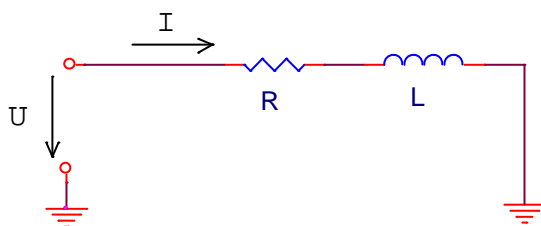
K vytvoření modelu řízení krokového motoru a jeho simulaci jsem zvolil software společnosti The MathWorks Inc. MATLAB 7.1 a jeho nadstavbu pro řešení diferenciálních rovnic SIMULINK.

MATLAB je výkonný, široce rozšířený nástroj pro technické výpočty a vizualizaci jejich výsledků. Nástroje MATLABu jsou členěny do tzv. *Toolboxů*, zejména kvůli rozlišení účelu použití a zpřehlednění pro uživatele. Pro řešení úloh s elektrickými obvody je v SIMULINKu možno použít *Power System Blockset*, který obsahuje většinu běžně používaných obvodových prvků.

Alternativně by se k řešení této úlohy mohl použít OrCad PSpice, který je však orientován výhradně na simulaci elektrických obvodů, zatímco MATLAB představuje komplexnější nástroj.

2.1 Náhradní obvod fáze motoru

Nejjednodušším náhradním obvodem vinutí motoru je sériový L-R obvod. Použití tohoto náhradního obvodu je možné pouze za značných zjednodušujících podmínek. Např. neuvažujeme rozptylovou reaktanci vinutí, ztráty v železe ani vzájemnou indukčnost vinutí. Protože simulování chování krokového motoru nebylo cílem této práce (nýbrž má sloužit pouze pro ilustraci a jako pomocný nástroj při návrhu elektronické řídicí jednotky motoru), zvolím jako náhradní obvod sériové zapojení L-R. Přesnější matematický popis chování krokového motoru uvádí [2].



Obrázek 2.1 Sériový L-R obvod

2.1.1 Sériový L-R obvod

Sériový L-R obvod tvoří dynamickou soustavu 1.řádu, kterou lze popsat pomocí časové konstanty τ a statického zesílení K .

$$K = \frac{U}{R} \quad (2.I)$$

Statické zesílení je dáno poměrem napájecího napětí U a odporu R , který vyjadřuje činný odpor vinutí. Statické zesílení určuje odezvu systému na buzení vstupním signálem po odeznění přechodových jevů – tedy teoreticky v čase $t \rightarrow \infty$.

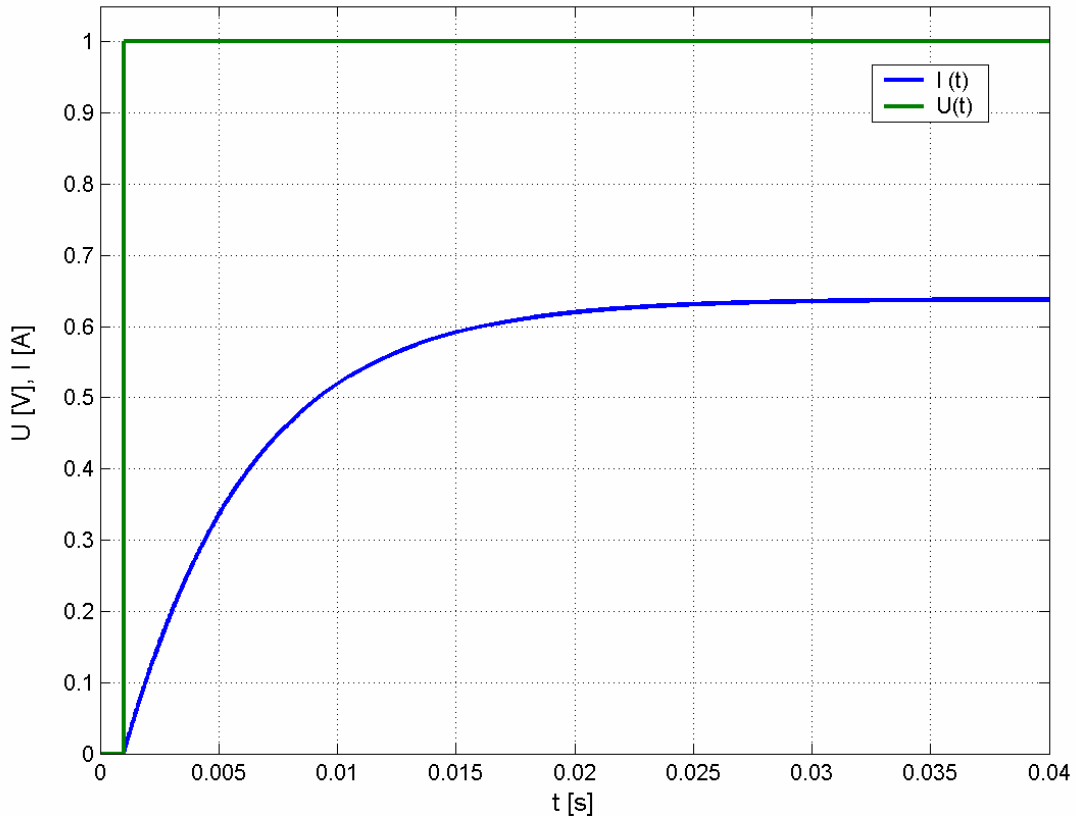
$$\tau = \frac{L}{R} \quad (2.II)$$

Časová konstanta τ vyjadřuje rychlost odezvy systému na buzení. V čase $t = \tau$ je hodnota proudu obvodem přibližně $I(\tau) = 0,632I(\infty)$. Tedy 63,2% hodnoty ustáleného proudu. Přechodový děj se většinou považuje za odezněný v době $t_s = (4 + 5)\tau$.

Odezva proudu na skokovou změnu napětí na vstupu má pak tvar

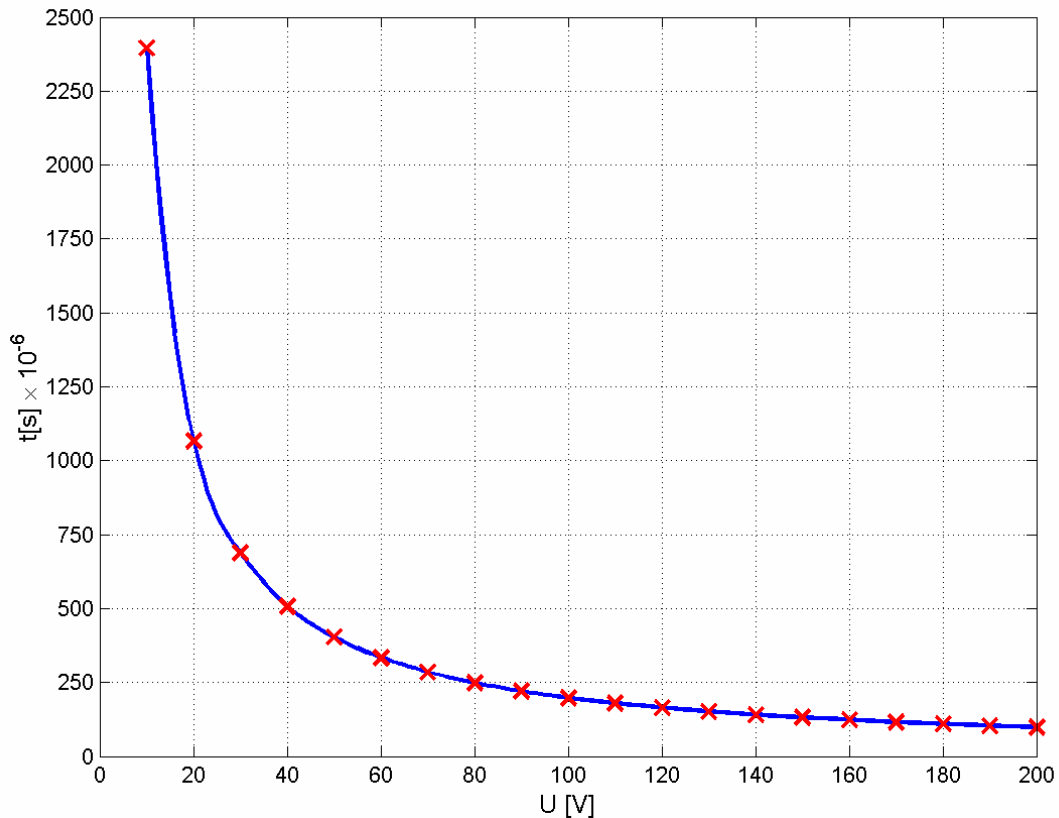
$$i(t) = K \left(1 - e^{-\frac{t}{\tau}} \right) \quad (2.III)$$

Významnou charakteristikou pro posuzování chování dynamických soustav je přechodová charakteristika. Jedná se o odezvu soustavy na buzení jednotkovým skokem.



Obrázek 2.2 Příklad přechodové charakteristiky pro hodnoty: $L = 8,4 \text{ mH}$, $R = 1,566 \Omega$.
Tedy $\tau = 5,364 \text{ ms}$, $K = 0,638$

Z přechodové charakteristiky je patrné, že pokud přivedeme na fázi motoru napětí $U = 1 \text{ V}$ bude ustálená hodnota proudu $I(\infty) = 0,638 \text{ A}$. Uvažujme jmenovitý proud fází motoru daný výrobcem $I_{jm} = 2,3 \text{ A}$. Pro dosažení jmenovité hodnoty proudu by pak stačilo přivést napětí $U = 3,6 \text{ V}$. Avšak je nutné počítat s časovou konstantou τ . Uvažujeme-li ustálení přechodového jevu za dobu $t_s = 4\tau$ a požadujeme-li dosažení jmenovitého proudu, pak maximální frekvence krokování je $f_k = \frac{1}{t_s} = 46,6 \text{ Hz}$. To je samozřejmě nedostatečné.



Obrázek 2.3 Závislost doby potřebné k dosažení jmenovitého proudu na napájecím napětí

Pro dosažení potřebné dynamiky pohonu je tedy nutné zvyšování napájecího napětí U , abychom dosáhli co možná nejkratší doby náběhu proudu na jmenovitou hodnotu. Zvyšujeme-li napětí impulsů přiváděných na motor klesá doba t nutná k dosažení jmenovitého proudu $I_{jm} = 2,3 \text{ A}$. Závislost této doby t na napětí U má hyperbolický charakter. Napětí nelze zvyšovat nad meze dané konstrukcí motoru (omezujícím faktorem je zejména mezizávitová izolace).

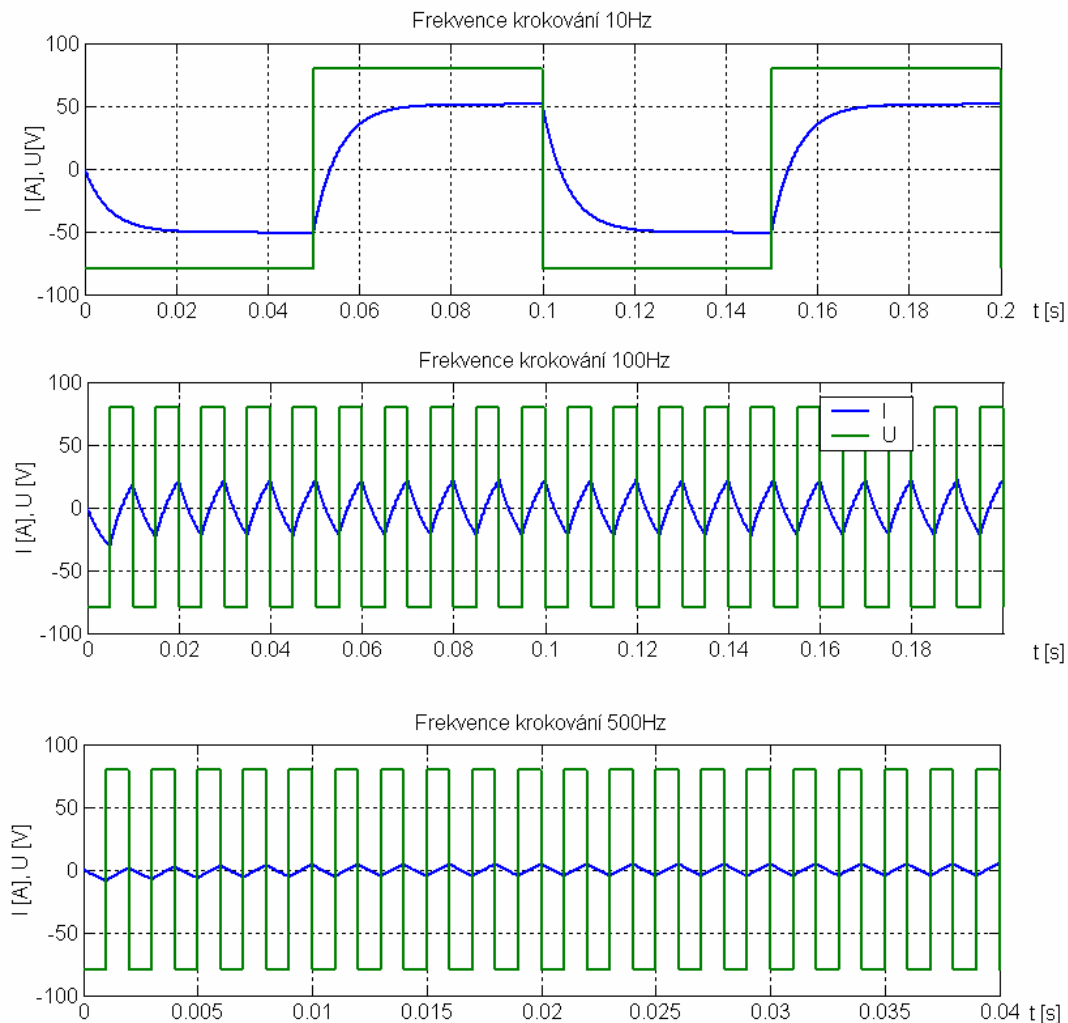
Z konstrukčních důvodů (maximální napětí spínacích součástek, kondenzátorů, dostupnost napájecího zdroje atd.) budu dále uvažovat maximální napájecí napětí $U = 80 \text{ V}$, při němž je doba potřebná k dosažení jmenovitého proudu $t = 243 \mu\text{s}$.

Při napájecím napětí $U = 80 \text{ V}$, jmenovitém odporu vinutí $R = 1,566 \Omega$ je statické zesílení soustavy $K = 51,08$ a tedy ustálený proud $I(\infty) = 51,08 \text{ A}$, což je pro uvažovaný motor s jmenovitým proudem $I_{jm} = 2,3 \text{ A}$ nepřípustná hodnota.

Je tedy patrné, že proud vinutím je třeba vhodným způsobem omezovat. Zřejmě pak zde proti sobě stojí protichůdné požadavky: zajistit dobrou dynamiku pohonu co nejrychlejším náběhem proudu vinutím, čehož dosáhneme patřičně vysokým napájecím napětím. Na druhé straně by však neměl být překračován jmenovitý proud motoru daný výrobcem z důvodu nadměrného oteplování motoru. Oteplení motoru nad hodnoty doporučené výrobcem způsobuje degradaci drážkové izolace, což má za následek výrazné snižování životnosti motoru.

2.1.2 Závislost velikosti a průběhu proudu na rychlosti krokování

Uvažujme vinutí stále stejných parametrů a bipolární způsob řízení. Se zvyšující se frekvencí krokování se bude stále více projevovat časová konstanta τ soustavy. Tím se dojde k výraznému snižování maximální hodnoty proudu vinutím. Od jisté frekvence krokování (cca 1000 kroků/s) již proud vinutím nedosáhne ani jmenovité hodnoty.



Obrázek 2.4 Závislost velikosti a průběhu proudu na rychlosti krokování

2.1.3 Závislost momentu motoru na proudu procházejícím vinutím

Opět za zjednodušujících podmínek můžeme pro krokový motor uvažovat stejné vztahy jako pro stejnosměrný motor s cizím buzením.

$$M_h(t) = C_s \Phi i(t) \quad (2.IV)$$

Hnací moment stejnosměrného motoru je úměrný konstantě stroje, magnetickému toku a proudu vinutím. Navíc lze magnetický tok považovat za konstantní $\Phi = konst.$, pak tedy

$$M_h(t) = ki(t) = M_z + J \frac{d\varpi}{dt}, \quad (2.V)$$

kde M_z je moment zátěže na hřídeli zvýšený o pasivní odpory motoru., J potom zahrnuje moment setrvačnosti samotného motoru a všechny momenty setrvačnosti pohybujících se částí přepočtené na hřídel motoru.

Hnací moment motoru je tedy v každém okamžiku přímo úměrný proudu procházejícímu vinutím. Je tedy jasné, že při vyšších frekvencích krokování dojde ke ztrátě točivého momentu a naopak maximum točivého momentu bude dostupné již od nulové rychlosti.

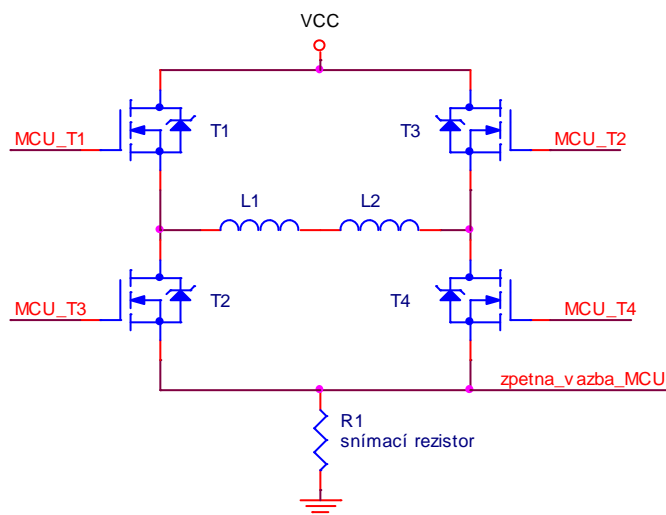
2.2 Možnosti řízení proudu vinutím

Na základě předchozích poznatků je jasné, že pro vývoj řídicí jednotky krokového motoru je potřeba zvolit vhodný způsob řízení proudu vinutím.

Zmíněné jsou pouze ty metody, které jsem při vývoji řídicí jednotky vyzkoušel na nepájivém poli, přičemž tyto se dle teoretických znalostí jevily pro daný účel jako nejvhodnější. Podrobněji se problematice věnuje např. [1].

2.2.1 Řízení proudu s pevnou dobou vypnutí spodního tranzistoru (current control with fixed sink switch off-time)

První metodou, kterou jsem vyzkoušel pro řízení velikosti proudu vinutím motoru je řízení s pevnou dobou vypnutí spodního tranzistoru. Tento způsob je založen na hlídání velikosti proudu ve vinutí pomocí snímacího rezistoru.



Obrázek 2.5 Principiální zapojení pro řízení proudu s pevnou dobou vypnutí

MOSFET tranzistory jsou zapojeny do H-můstku a jsou přes přizpůsobovací obvody ovládány jednočipovým mikropočítačem. Rezistor R_I , slouží ke snímání velikosti proudu protékajícího vinutím. Z Ohmova zákona plyne, že úbytek napětí na rezistoru je roven $U_{RI} = R_I I$, toto napětí je přivedeno na vstup A/D převodníku mikropočítače a tím je vytvořena zpětná vazba.

Sepneme-li např. tranzistory T_1, T_4 , dojde dle vztahu (2.III) k exponenciálnímu růstu proudu I vinutím, dosáhne-li napětí U_{RI} přednastavené hodnoty (odpovídající jmenovitému proudu vinutím), jednočip vypíná tranzistor T_4 na pevně nastavenou dobu (off-time). Vlivem indukčnosti vinutí se indukuje napětí opačné polarity a to otevírá diodu „integrovanou“¹ v tranzistoru T_3 . Tím se vinutí zkratuje a proud exponenciálně klesá.

Po uplynutí doby off-time je znovu sepnut tranzistor T_4 a proud vinutím opět vzrůstá. Během jednoho kroku se výše popsaná sekvence opakuje a tím je omezena maximální hodnota proudu.

¹ Nechtěný důsledek technologie výroby. Struktura tranzistoru se ve směru source-drain chová jako dioda.

Jedná se o poměrně jednoduchý způsob řízení, hlavním nastavovaným parametrem je doba *off-time*, která ovlivňuje velikost zvlnění proudu označované I_{ripple} . Problémem je nastavení doby *off-time* při použití tohoto řízení pro motory s různou indukčností.

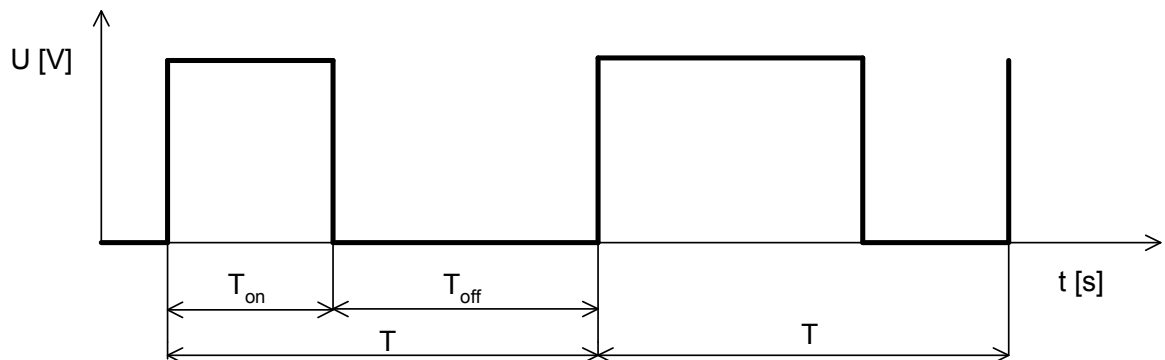
Neexistuje univerzální nastavení a při změně použitého motoru by bylo nutné řídicí jednotku přeprogramovat s novým nastavením *off-time*. Více informací o tomto řízení viz [1].

2.2.2 Řízení proudu vinutím pomocí pulsně-šířkové modulace (PWM)

PWM neboli pulsně-šířková modulace je velmi rozšířeným typem modulace používaným např. pro řízení výkonu zařízení pracujících se stejnosměrným napájením (SS motory), v DC-AC střídačích, ale také ke kódování signálu.

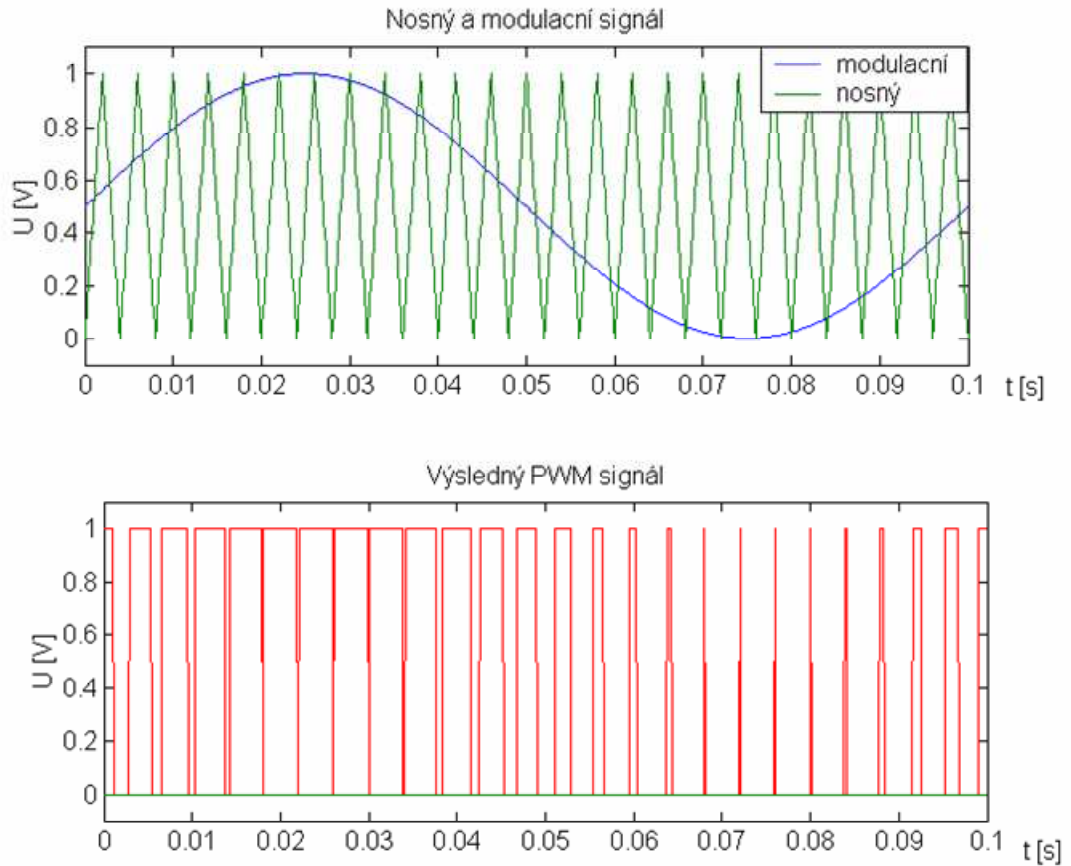
PWM je založena na změně střídý obdélníkového průběhu v závislosti na modulačním signálu. Střídou rozumíme poměr $D = \frac{T_{on}}{T_{off}}$. Frekvence PWM signálu

$$f = \frac{1}{T} = \frac{1}{T_{off} + T_{on}} = konst.$$



Obrázek 2.6 Popis časových úseků PWM signálu

Nejjednodušším způsobem generování PWM signálu je komparování trojúhelníkového nosného signálu a signálu modulačního, přičemž by mělo platit

$$f_{\text{nosný}} \geq 20 * f_{\text{modul}}$$


Obrázek 2.7 Příklad generování PWM signálu

Tohoto způsobu modulace lze velmi výhodně využít k řízení proudu vinutím krokového motoru.

Základní zapojení je shodné viz Obrázek 2.5. K bipolárnímu řízení jednoho plného H-můstku postačí jeden PWM generátor, jehož signál však musí být střídavě (dle aktuálního kroku – viz Obrázek 1.5) přiváděn na spodní tranzistory (dále low-side) T_2, T_4 . Základem modulačního signálu pro PWM generátor je upravené napětí U_{RI} snímacího rezistoru R_I . Tedy zpětnovazební signál.

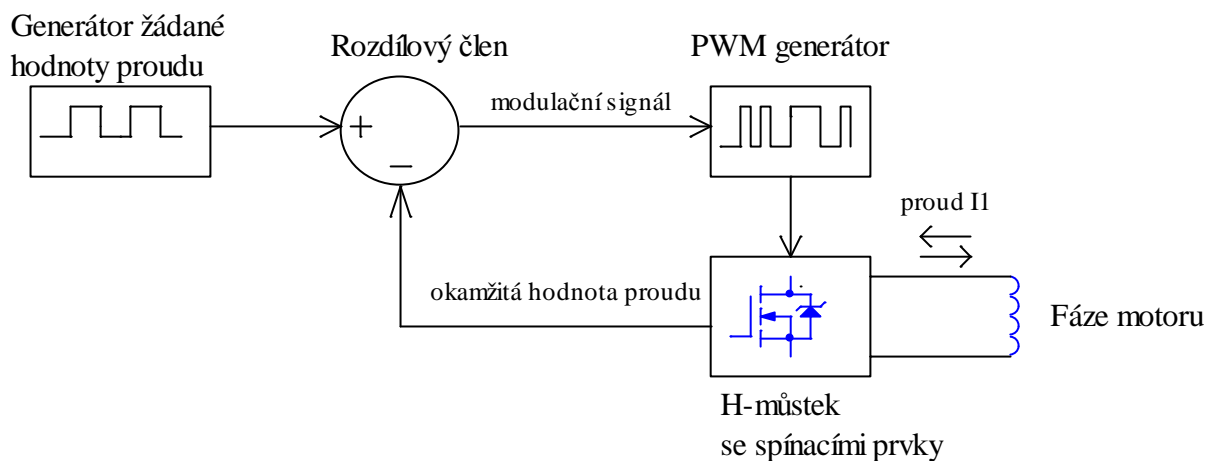
Myšlenka řízení pomocí PWM je podobná předchozímu popsanému typu řízení, kdy se k požadovanému poklesu proudu opět využívá zkratování vinutí přes diodu v horním tranzistoru T_1 nebo T_3 ke kterému dojde při rozepnutí low-side tranzistoru T_2 nebo T_4 (dle aktuálního směru proudu zátěží). Rozdíl je pouze v tom, že doba

vypnutí low-side tranzistoru (nazvaná *off-time* v předchozím případě) se při PWM řízení mění. Velikost a charakter těchto změn je dán nastavením regulátoru – softwarem mikrokontroléru.

Pomocí PWM modulace lze dosáhnout lepších výsledků při řízení proudu než pomocí *fixed sink switch off-time* metody. Navíc lze při použití PWM jednodušeji realizovat mikrokrokování. Tento způsob řízení je však složitější a vyžaduje buď dva PWM generátory pro jeden H-můstek nebo realizaci elektronického přepínání signálu z jednoho PWM generátoru.

2.3 Simulační model

Na základě teoretického rozboru jsem sestavil jednoduchý simulační model dvoufázového krokového motoru s dvoufázovým bipolárním řízením a PWM řízením proudu vinutím.

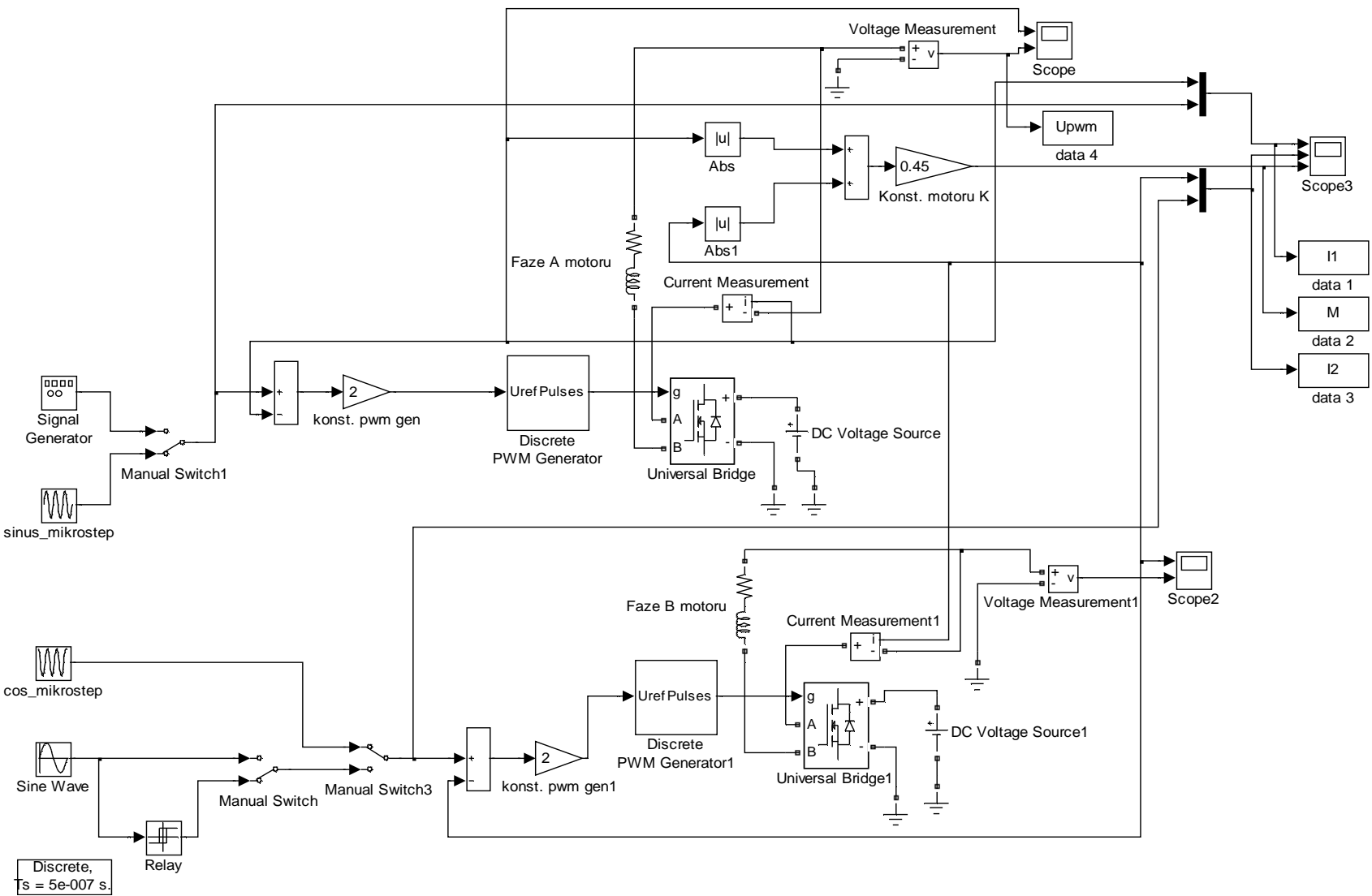


Obrázek 2.8 Blokové schéma simulačního modelu

Na výše uvedeném blokovém schématu lze dobře popsat simulační model. Generátor žádané hodnoty vytváří řídicí veličinu regulovaného obvodu. Pomocí přepínačů *Manual switch* (viz Obrázek 2.9) lze vybrat tvar požadovaného proudu – obdélníkový průběh, sinusoida nebo signál pro čtyřnásobné mikrokrokování.

Rozdílový člen porovnává okamžitou a požadovanou hodnotu proudu a jeho výstupem je modulační signál pro PWM generátor. Ten ovládá MOSFET tranzistory v H-můstku. Zpětnou vazbu tvoří signál okamžité hodnoty proudu a je jím ovlivňován modulační signál PWM generátoru.

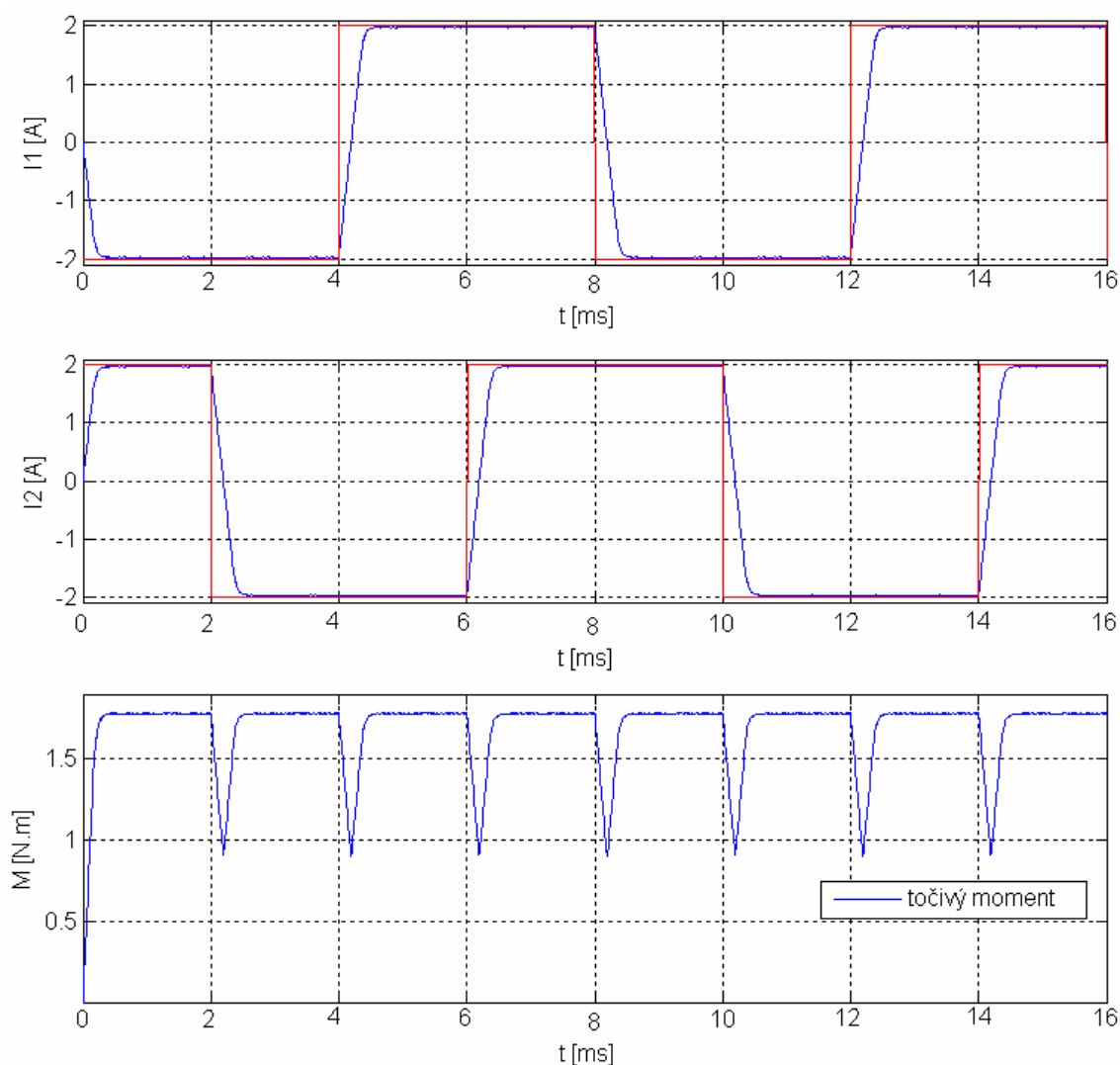
Pro obě fáze motoru je vytvořen výše popsáný model a z proudů I_1 , I_2 procházejících oběma fázemi je dle vztahu (2.V) vyhodnocena okamžitá hodnota točivého momentu.



Obrázek 2.9 Schéma simulačního modelu

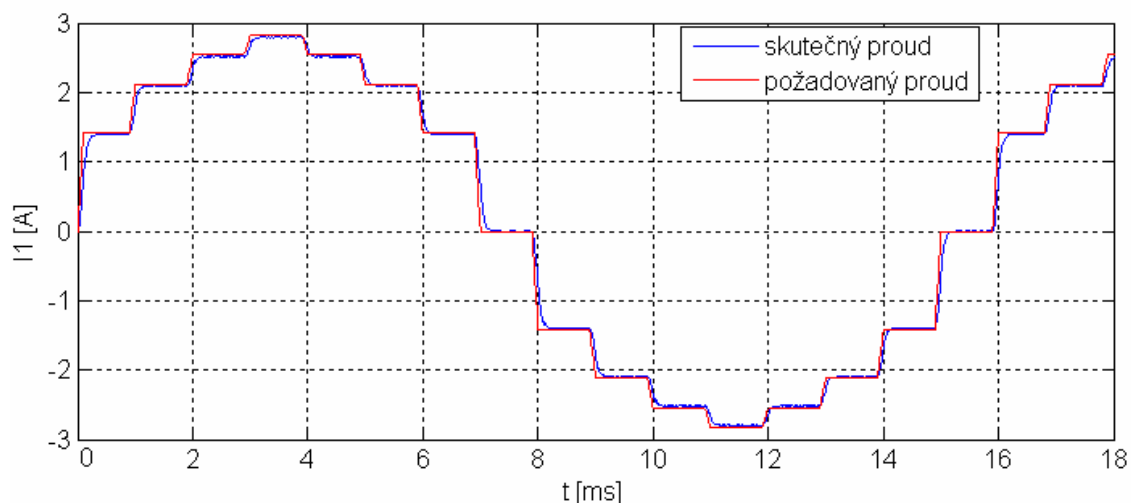
2.4 Výsledky simulací

Na simulačním modelu jsem vyzkoušel zejména závislost průběhu proudu motorem a momentu motoru na rychlosti krokování a napájecím napětí. Navržené řízení proudu vinutím pomocí PWM se dle simulací ukazuje jako funkční. Rovněž jsem vyzkoušel chování modelu při mikrokrokování.



Obrázek 2.10 Proudů vinutím a výsledný točivý moment motoru

Obrázek 2.10 ukazuje průběhy proudů vinutími a výsledný točivý moment při požadavku obdélníkového průběhu proudu. Za povšimnutí stojí průběh točivého momentu motoru. Protože je ve zjednodušeném modelu točivý moment počítán dle vztahu (2.V) je zřejmé, že při změně směru toku proudu vinutím se projeví její konečná rychlost. A to jako mžikový pokles točivého momentu.



Obrázek 2.11 Průběh proudu vinutím při mikrokrokování

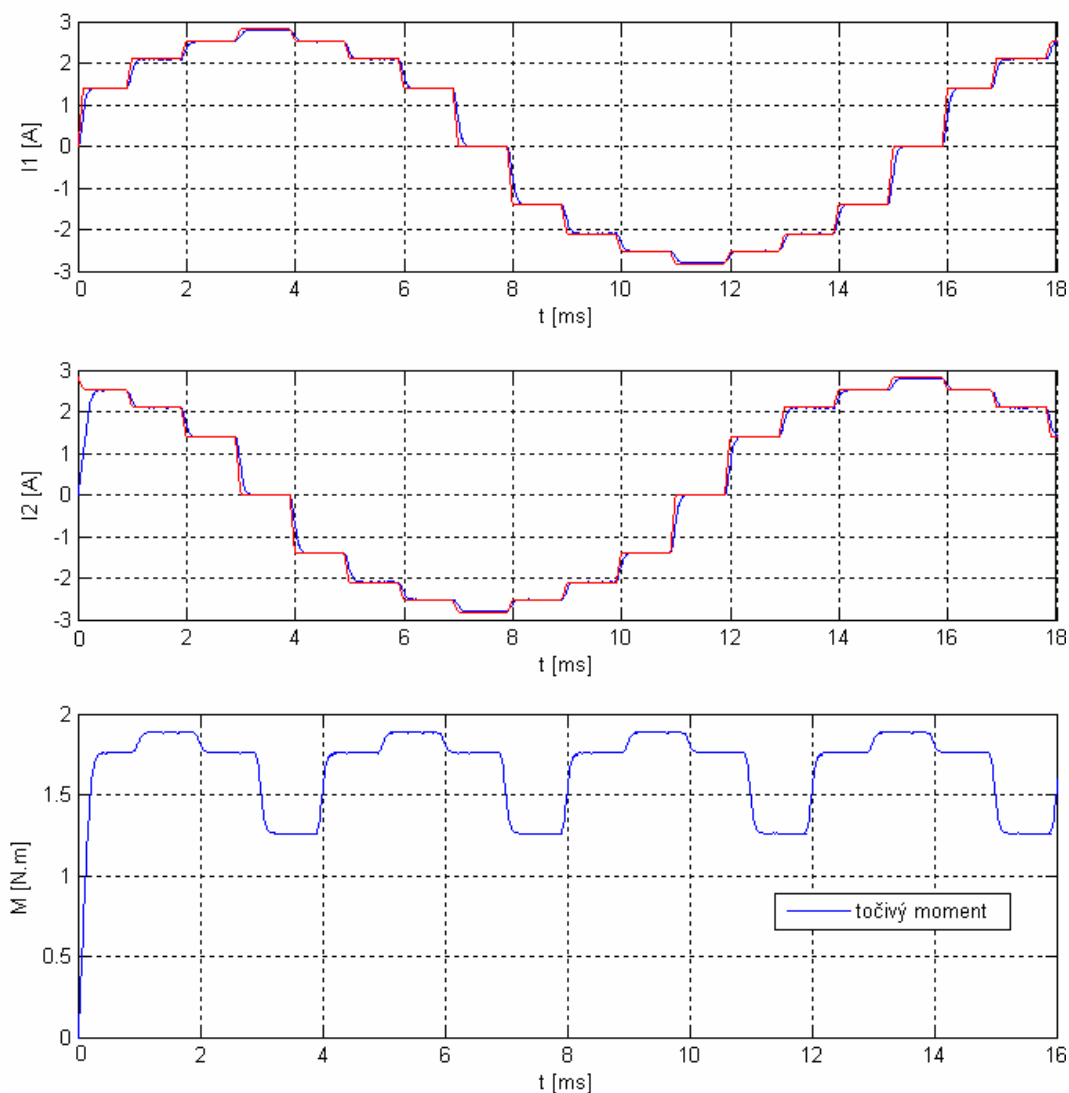
Je patrné, že proud vinutím by se tím více blížil ideální sinusoidě, čím jemnější mikrokrokování by bylo použito. To ale klade vyšší nároky na rychlost řídicích obvodů i spínacích součástek. Rozumným kompromisem se jeví použití čtyřnásobného mikrokrokování².

Porovnáme-li průběh proudu vinutím při celo- a mikrokrokování, pak je na první pohled zřejmá odlišnost v rychlosti změny proudu vinutím (při stejné rychlosti krokování). Zatímco při celokrokování je $\frac{di}{dt} \approx 5 \text{ A/ms}$, pak při mikrokrokování je

$\frac{di}{dt} \approx 0,6 \text{ A/ms}$. To je hlavní příčina výhod a odlišností mikrokrokování oproti způsobu

řízení s obdélníkovým průběhem proudu zmíněných v kapitole 1.6.

² Průmyslové jednotky často nabízejí osmi i vícenásobné mikrokrokování



Obrázek 2.12 Průběh proudu vinutím a moment motoru při mikrokrokování

Pro možnost porovnání točivého momentu při celokrokování a mikrokrokování jsem nastavil maximální proud vinutím $I_{max} = 2,828 \text{ A}$, což znamená efektivní hodnotu proudu $I_{ef} = 2 \text{ A}$ za předpokladu, že považujeme průběh proudu vinutím při mikrokrokování za sinusový. Průběh točivého momentu je mírně odlišný než při celokrokování.

Lze říci, že točivý moment motoru má při mikrokrokování výhodnější charakter, protože jeho „propady“ jsou menší zhruba o 10% a nejsou tak strmé.

2.5 Závěry ze simulací a teoretických poznatků

Nejdůležitějším bodem této části práce je přenesení výsledků simulací a teoretického rozboru do vlastní praktické realizace systému. Je proto potřeba porovnat požadavky na realizovaný systém a dle nabytých teoretických poznatků zvolit nejvhodnější způsob řízení motorů.

2.5.1 Použité krokové motory a volba typu řízení

Pro pohon všech os polohovacího systému budou použity krokové motory výrobce, firmy MICROCON s.r.o. typ: SM2321-1400 Tyto motory byly na Katedru elektrotechniky zakoupeny již v minulosti.

Hlavní parametry krokového motoru SM2321-1400:

- velikost kroku 1,8°
- jmenovitý proud 2,3/4,6 A (sériové/paralelní zapojení)
- indukčnost 8,4/2,1 mH
- odpor vinutí 1,52/0,38 Ω
- točivý moment při 500 Hz 1,1 N.m

Vinutí motoru jsou vyvedena viz Obrázek 1.3. Z důvodu využití stávajícího napájecího zdroje je kvůli velikosti odebíraného proudu nutné použít sériové zapojení vinutí motoru. Pro splnění požadavku dostatečného točivého momentu výsledného pohonu nelze při sériovém zapojení vinutí motoru uvažovat o jiném než bipolárním, dvoufázovém řízení.

Pro řízení velikosti proudu vinutím zvolím PWM modulaci, kterou lze poměrně snadno realizovat pomocí jednočipového mikrokontroléru.

Výkres a fotografie použitého motoru jsou součástí přílohy [C].

2.5.2 Problémy s rezonancí motoru

Při vývoji a zkoušení řídicí jednotky krokového motoru se vyskytly poměrně výrazné potíže s rezonancí. Při rychlosti krokování blízké vlastní rezonanční frekvenci docházelo k tak významnému poklesu točivého momentu, že se rotor motoru téměř zastavoval, nebo byl jeho chod výrazně nepravidelný a doprovázený značnými zvukovými „efekty“. Navíc se problém s rezonancí v menším rozsahu vyskytoval i pokud frekvence krokování byla celočíselným násobkem vlastní rez. frekvence motoru. Bylo proto nutné získat další teoretické poznatky a pokusit se nalézt vhodné řešení.

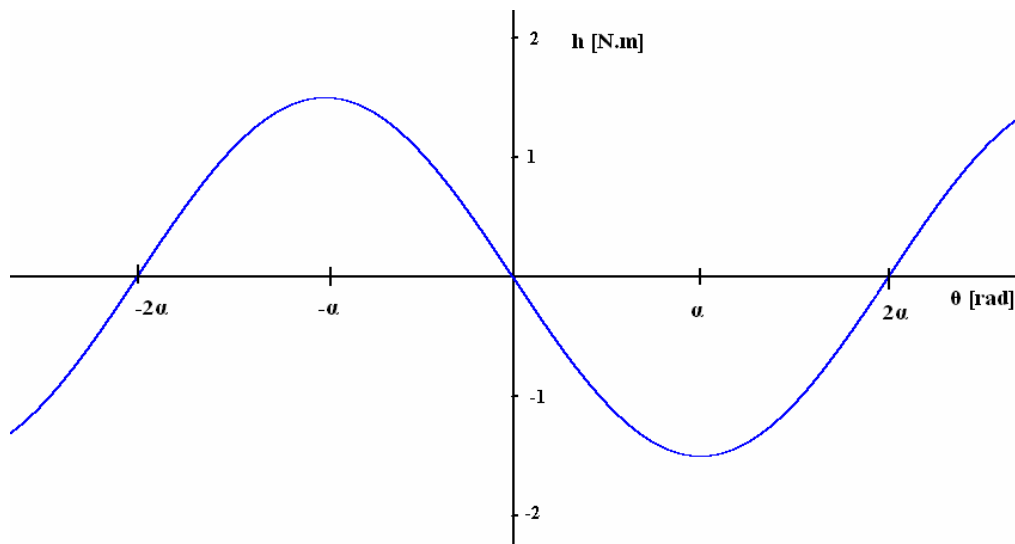
Rezonance je u mechanických systémů definována jako děj, který nastane, pokud je frekvence vnucených kmitů systému shodná s vlastní rezonanční frekvencí, která je

$$\omega_0 = \sqrt{\frac{k}{m}}, \quad (2.VI)$$

kde k je konstanta tuhosti a m hmotnost. Pro rotaci můžeme vztah upravit.

$$\omega_0 = \sqrt{\frac{k}{J}} \quad (2.VII)$$

V případě krokového motoru, kde budeme předpokládat sinusovou závislost statického momentu na poloze hřídele rotoru viz Obrázek 2.13, lze odvodit vzorec (2.VIII)



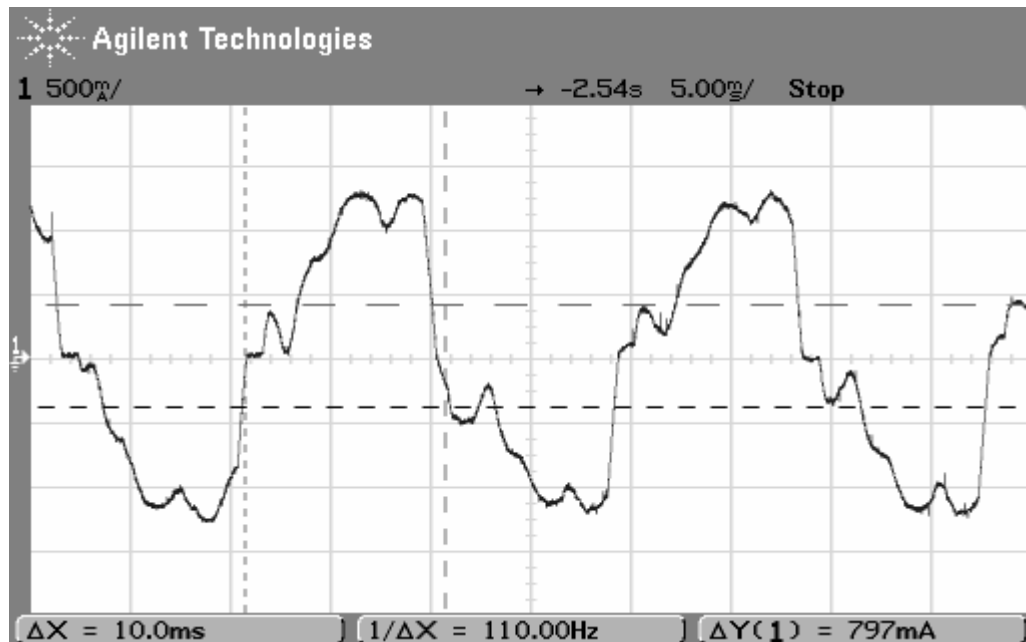
Obrázek 2.13 Průběh statického momentu

$$f_{rez} = \sqrt{\frac{h}{8\pi J_r \alpha}} \quad (2.VIII)$$

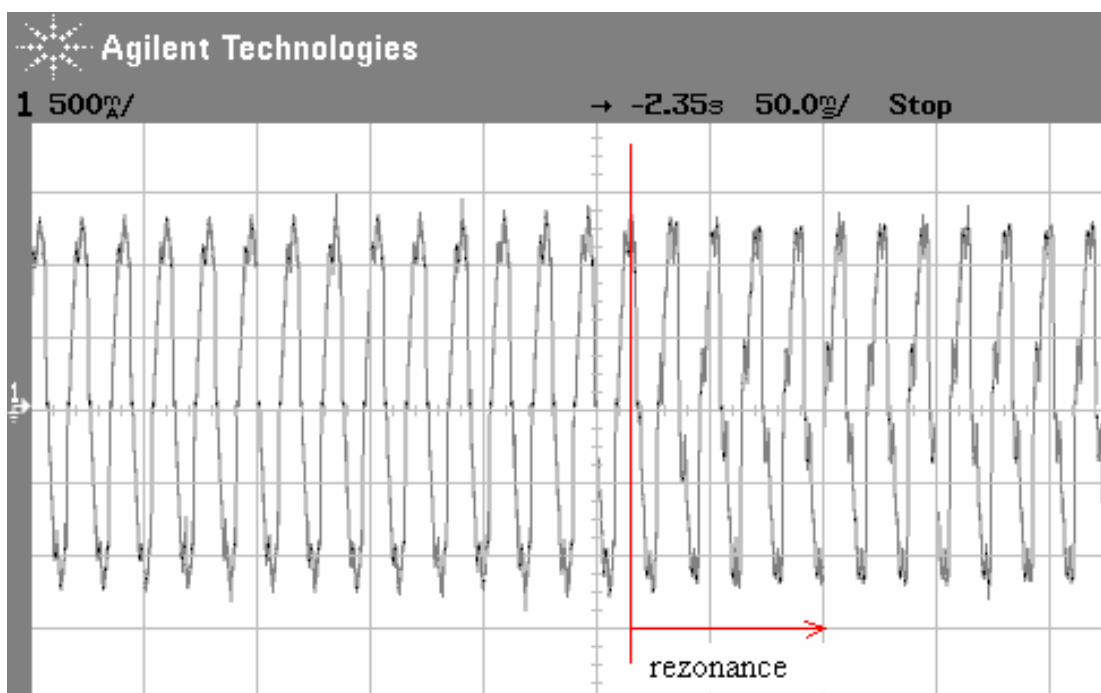
Dosažením údajů z katalogového listu výrobce do vztahu (2.VIII) dostáváme $f_{rez} = 275,6 \text{ Hz}$. Vypočtená f_{rez} se rozchází s experimentálně zjištěnou hodnotou $f'_{rez} \approx 220 \text{ Hz}$. Tento rozdíl může být způsoben nepřesnými údaji výrobce. Moment setrvačnosti rotoru udaný výrobcem $J_r = 0,025 \cdot 10^{-3} \text{ kg} \cdot \text{m}^2$, tedy velmi malá hodnota, která mohla být změřena s podstatnou nejistotou měření, což zřejmě může způsobovat rozdíl mezi vypočtenou a naměřenou hodnotou f_{rez} . Odvození vzorce (2.VIII) a další podrobnosti lze nalézt v [1].

Průběh proudu při rezonanci

Obrázek 2.14 znázorňuje průběh proudu jednou fází motoru při mikrokrokování, přičemž frekvence krokování $f_k \approx f_{rez} \approx 220 \text{ Hz}$. Porovnáme-li tento průběh s průběhem při $f_k = 100 \text{ Hz}$ (viz Obrázek 1.7) vidíme, že dochází k výrazné „deformaci“ procházejícího proudu.

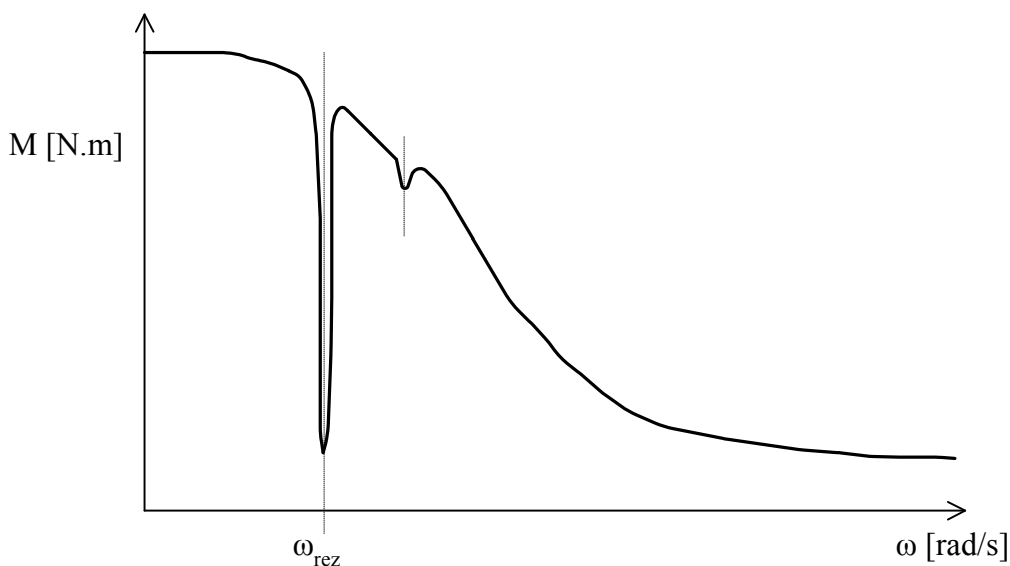


Obrázek 2.14 Průběh proudu při rezonanci



Obrázek 2.15 Změny v průběhu proudu při zvyšování f_k

Obrázek 2.15 zachycuje změnu v průběhu proudu při plynulém zvyšování f_k . Opět je tato změna patrná, když se f_k dostává do oblasti vlastní rezonanční frekvence motoru.



Obrázek 2.16 Příklad momentové charakteristiky dvoufázového krokového motoru s bifilárním vinutím zapojeným do série (při celokrokování)

V momentové charakteristice je patrný velmi výrazný pokles točivého momentu pokud se rotor otáčí úhlovou rychlostí $\omega = \omega_{\text{rez}}$.

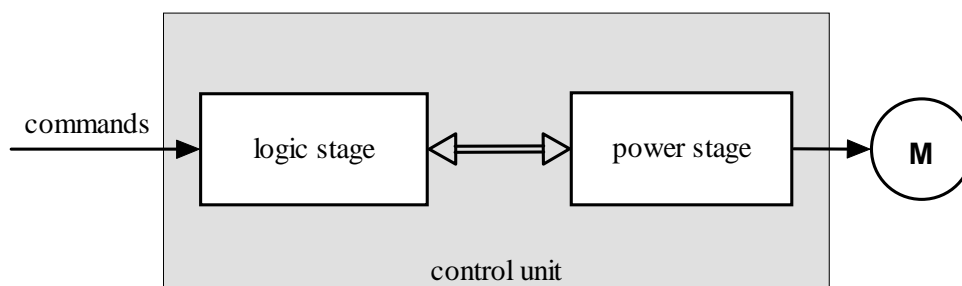
Nejjednodušším řešením těchto potíží by patrně bylo vyvarovat se otáčení zmíněnou rychlostí ω_{rez} . Ovšem s přihlédnutím k požadavku lineárního řízení rychlosti otáčení a k tomu, že se problém s rezonancí vyskytoval i na násobcích ω_{rez} , je nutné toto řešení zavrhnout.

Mnohem reálnějším východiskem se ukázalo použití mikrokrokování. Jak je patrné viz Obrázek 2.14 i při mikrokrokování má rezonance vliv na průběh proudu motorem avšak nedochází ke zdaleka tak výraznému „propadu“ točivého momentu v oblasti vlastní rezonanční frekvence motoru a tím se vyvarujeme ztráty kroků při pohybu.

Z pohledu úprav, které bylo nutné provést, byl tento způsob řízení též přijatelný. V zásadě se totiž jednalo pouze o změny v řídicím programu mikrokontroléru, nikoli ve vlastním hardware řídicí jednotky.

3 Realizace řídicí jednotky krokových motorů

Po provedení simulací a volbě typu řízení bylo možné přistoupit k realizaci řídicí jednotky. Rozhodl jsem se pro řešení se dvěma mechanicky oddělenými stupni. Prvním je logická, řídicí část a druhým výkonová. Výhodou tohoto řešení je snazší orientace v celé jednotce, jednodušší hledání případných závad a při eventuální nutnosti opravy možnost vyměnit pouze příslušnou část. Naopak nevýhodou byla nutnost vyřešit propojení obou desek a složitější prostorové uspořádání. Kompletní schéma zapojení jednotky je součástí přílohy [A], fotografie pak součástí přílohy [C].



Obrázek 3.1 Návrh koncepce řídicí jednotky

3.1 Logická část jednotky

Prvním důležitým krokem návrhu bylo zvolit vhodný mikrokontrolér, který bude zajišťovat většinu funkcí výsledné jednotky – řídit výkonovou část, zpracovávat signály zpětné vazby a komunikovat s okolím.

3.1.1 Mikrokontrolér

Požadavky na mikrokontrolér:

- Dvojice 8bitových A/D převodníků.
- Dvojice 8bitových časovačů umožňujících generování PWM signálu.
- Alespoň jeden 16bitový časovač.
- JTAG rozhraní pro snadné programování a ladění aplikace.
- SPI rozhraní ke komunikaci s nadřazenou jednotkou.
- Dostatečné množství vstupně/výstupních portů.

- Dostatečný výpočetní výkon a taktovací frekvence.

V současné době je na trhu mnoho výrobců, jejichž mikrokontroléry výše zmíněné požadavky splňují např. Motorola, NEC electronics, Infineon a další. Já jsem po dohodě s vedoucím práce zvolil mikrokontrolér firmy ATMEL a to z důvodů mých předchozích zkušeností s těmito procesory, jejich snadné dostupnosti v ČR, jejich ceně a dostatečnému vybavení periferními obvody. Posledním, neméně důležitým důvodem, proč zvolit právě procesory této firmy byl fakt, že se na katedře již delší dobu zvažoval nákup profesionálního programátoru JTAGICE mkII a tento programátor se i posléze podařilo zakoupit.

Zvolena byla řada procesorů ATmega, konkrétně pak ATmega16, což představuje kompromis mezi cenou (cca 80 Kč), vybaveností pamětí a množstvím periférií.

Hlavní parametry 8bitového mikrokontroléru ATmega16:

- Architektura RISC, harwardská koncepce, 32 registrů, vykonání jedné instrukce v jednom hodinovém taktu, maximální taktovací frekvence 16 MHz, výkon 1 MIPS/MHz
- 16K Byte paměti FLASH pro program, 1K Byte SRAM
- 8×10bit A/D převodník, 2×8 bit, 1×16 bit časovače, 4×PWM
- programování v cílové aplikaci (ISP) přes SPI nebo JTAG rozhraní
- pracovní napětí 4,5–5,5 V, pouzdro DIP40/TQFP44
- struktura mikrokontroléru optimalizována pro programování v jazyce C

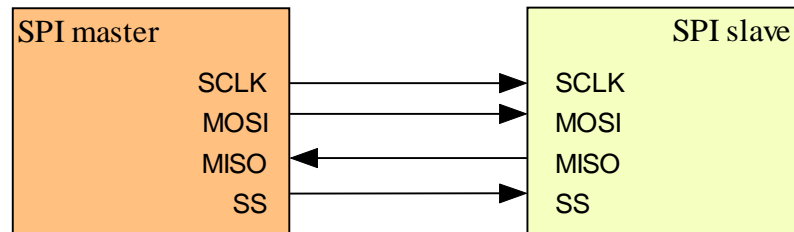
3.1.2 Komunikace mikrokontroléru s okolím

Logickou část jednotky lze programovat přes SPI nebo JTAG rozhraní s využitím ISP³ – tedy programování přímo ve výsledné aplikaci. Tento způsob má velkou výhodu ve velmi snadné změně programu procesoru přímo na cílové desce plošných spojů – procesor tedy není nutné umísťovat do patice a jakkoli s ním manipulovat při případné změně programu.

³ In system programming

SPI rozhraní (Serial peripheral interface)

SPI je jednoduché synchronní sériové rozhraní navržené firmou Motorola pro použití v embedded systémech. SPI využívá principu master–slave, přičemž komunikace mezi jednotkami je plně duplexní.

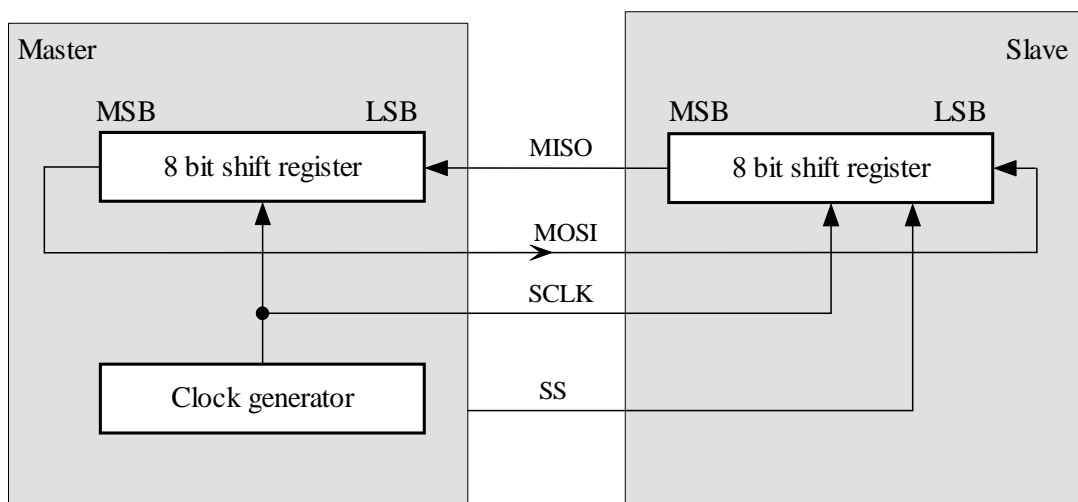


Obrázek 3.2 Propojení jednotek na SPI sběrnici (single master/single slave)

Významy jednotlivých signálů:

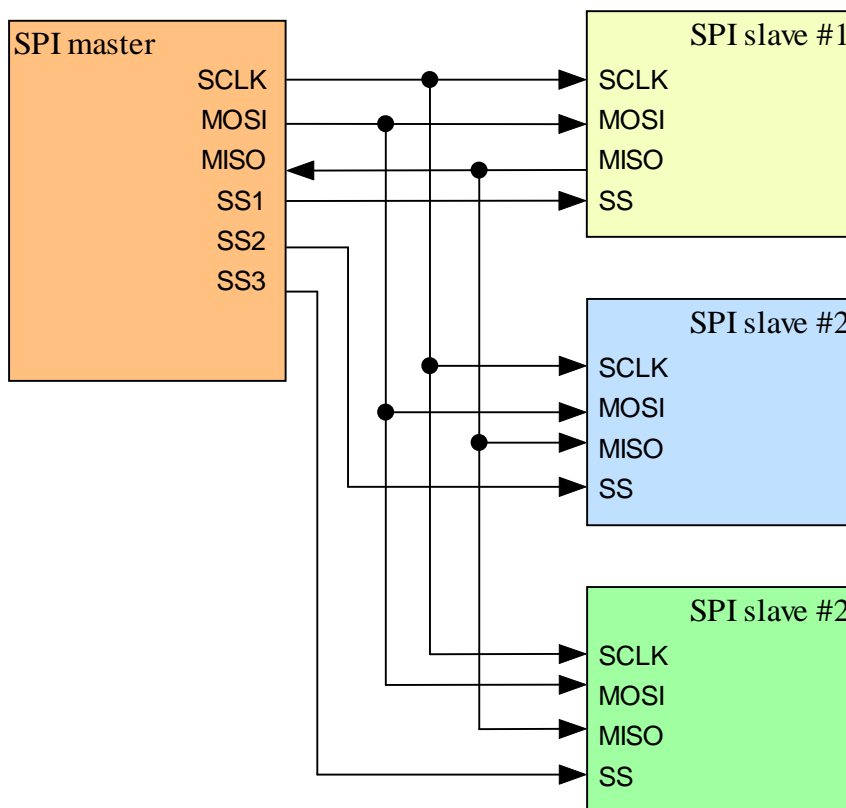
- SCLK: hodinový taktovací signál.
- MOSI: master out, slave in. Výstup dat z jednotky master, vstup dat do slave
- MISO: master in, slave out. Vstup dat do jednotky master, výstup ze slave
- SS: slave select. Tento signál slouží k vybrání příslušné slave jednotky. Má význam pouze při multi-slave konfiguraci.

Principiálně si lze přenos dat po SPI představit jako posun bitů v 16bitovém kruhovém posuvném registru v němž prvních osm bitů náleží jednotce master a zbylých osm jednotce slave.



Obrázek 3.3 Princip SPI komunikace

Zařízení master vždy generuje hodinový signál a pomocí pinu SS vybírá příslušnou slave jednotku.



Obrázek 3.4 Zapojení SPI sběrnice při multi-slave konfiguraci

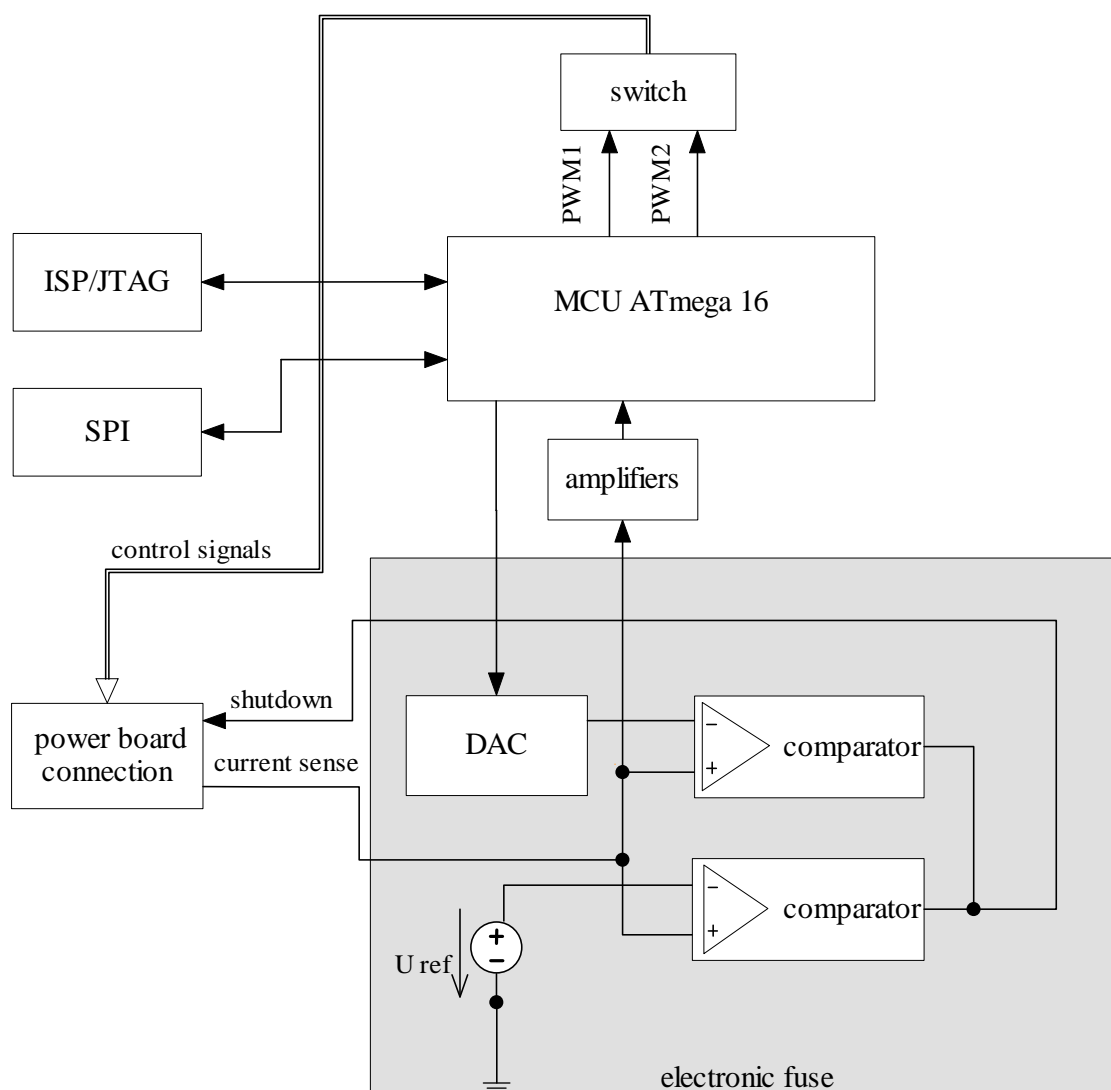
V návrhu řídicí jednotky krok. motoru počítám s využitím SPI zejména pro komunikaci s nadřazenou řídicí jednotkou viz Obrázek 3.4, přes toto rozhraní lze jednotky i programovat avšak k tomuto účelu bude více využíváno rozhraní JTAG, které umožňuje tzv. ISD (In system debugging) neboli ladění přímo v aplikaci. Více o SPI viz [4].

JTAG

JTAG je systém pro komunikaci a testování embedded zařízení, definovaný standardem IEEE 1149.1. Využívá metody skenování *Boundary Scan*. Tento systém byl zaveden v roce 1990.

V mnou navrhovaných řídicích jednotkách je JTAG použit k programování a ladění aplikace. V takovém případě je JTAG využíván zejména jako „transportní mechanismus“ umožňující přístup k debug-modulu mikrokontroléru. Více o JTAG viz [5].

3.1.3 Blokové schéma logické části jednotky

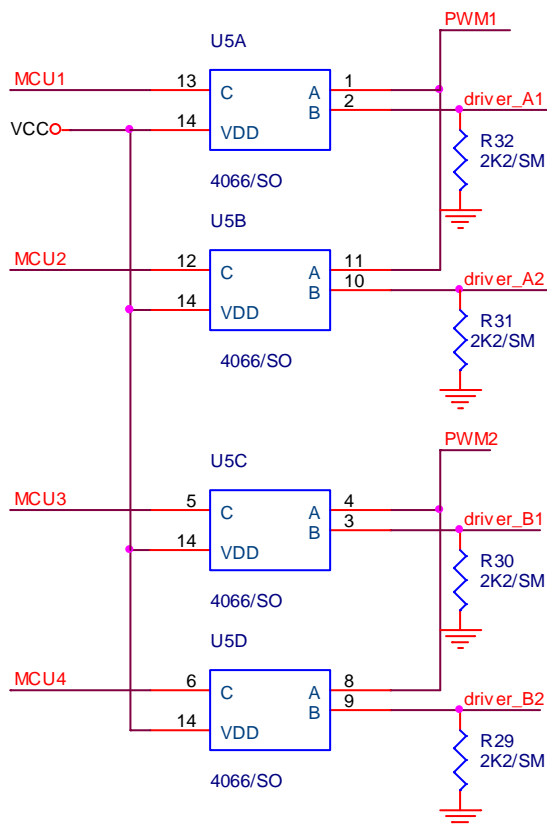


Obrázek 3.5 Blokové schéma logické části jednotky

Princip činnosti

Hlavním prvkem logické části jednotky je mikrokontrolér Atmega 16, který generuje řídicí signály pro spínání tranzistorů ve výkonové části jednotky. Na základě signálu zpětné vazby (current sense) mikrokontrolér mění střidu PWM signálu, kterým jsou ovládány tranzistory. Tím je regulován proud vinutím motoru na požadovanou hodnotu. Napěťový signál zpětné vazby nesoucí informaci o protékajícím proudu vinutím je 10× zesílen pomocí operačních zesilovačů kvůli lepšímu využití rozsahu A/D převodníku.

Protože signál dvou PWM generátorů je potřeba přivádět ke čtyřem různým dvojícím low-side tranzistorů H-můstku, realizoval jsem zapojení elektronického přepínače. Integrovaný obvod 4066 je elektronický přepínač analogových nebo digitálních signálů. Vstupem jsou signály PWM generátoru, které jsou v závislosti na stavech řídicích signálů z mikrokontroléru přiváděny na příslušný výstup. Na jednotlivé výstupy obvodu 4066 byly proti zemi připojeny rezistory, které slouží k rychlejšímu vybíjení vnitřních kapacit obvodu při přechodu ze stavu „1-sepnuto“ do stavu „0-rozepnuto“.



Obrázek 3.6 Zapojení elektronického přepínače

Elektronická pojistka

Poslední významnou částí logiky jednotky je blok electronic-fuse tedy elektronická pojistka. Požadavkem bylo realizovat tři signály, které umožní okamžité vypnutí proudu motorem. Prvním je signál získaný komparací hodnoty napětí zpětnovazebního signálu s napětím odporového děliče. Uživatel nastaví odporovým trimrem maximální přípustnou hodnotu proudu vinutím motoru a její překročení vyhodnotí komparátor s operačním zesilovačem.

Druhým zdrojem referenčního napětí pro komparaci se zpětnovazebním signálem je D/A převodník MCP4921, což je 12bitový převodník se sériovým vstupem dat. Rozhraní pro komunikaci s převodníkem se podobá SPI. Pro nastavení výstupního napětí jsou převodníku zaslány mikrokontrolérem dva datové bajty. První 4 bity jsou konfigurační a následných 12 bitů je interpretováno jako požadované napětí na výstupu.

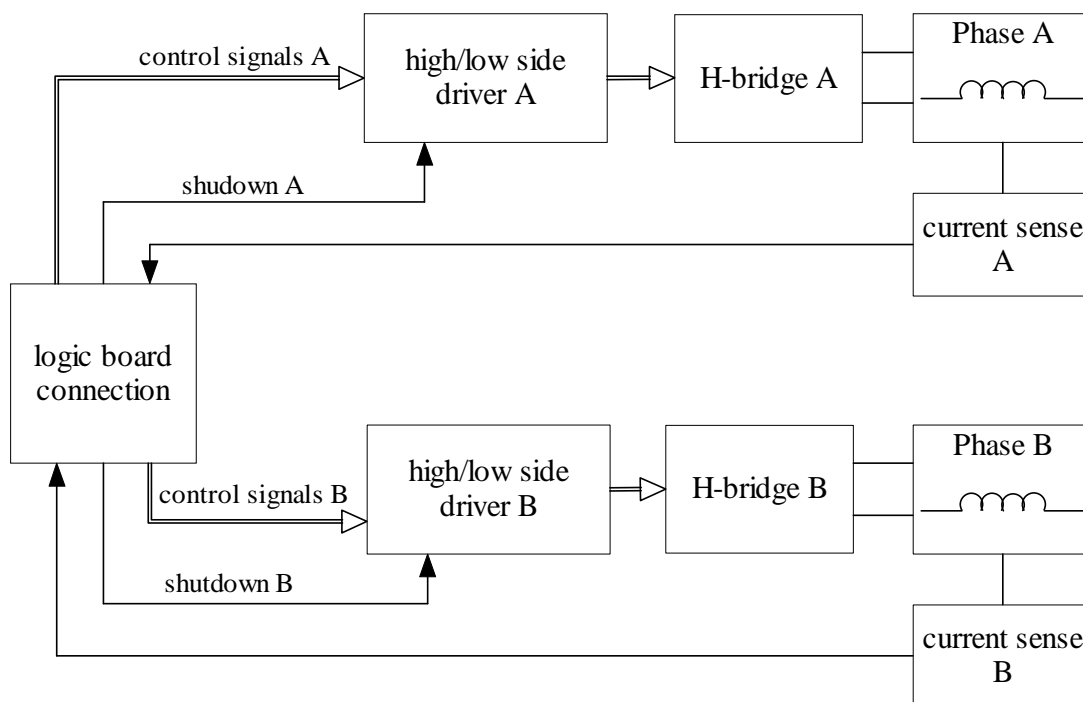
Výstupní napětí se počítá dle vztahu: $U_{out} = U_{ref} \frac{D}{2^{12}}$, kde D je dekadická hodnota zmíněných 12 bitů. Tímto způsobem získáváme možnost softwarového nastavení maximální přípustné hodnoty proudu.

Třetím signálem, umožňujícím vypnutí proudu motorem je signál přímo z I/O portu mikrokontroléru. Ten lze využít např. tehdy, pokud příslušný nadřazený systém vyhodnotí jakýkoli poruchový stav a předá informaci řídicí jednotce motoru, ta pak může okamžitě vypnout (odpojit) ovládaný motor.

Všechny tři výsledné signály elektronické pojistky jsou pomocí logického součtu, realizovaného diodovou logikou, převedeny v jeden signál, který ovládá *shutdown* vstup budičů tranzistorů ve výkonové části jednotky.

3.2 Výkonová část jednotky

Výkonová část řídicí jednotky je založena na *high and low side* budičích MOSFET tranzistorů IR2112 (výrobce IR⁴). Tyto budiče jsou řízeny spínacími signály z logické části jednotky přivedenými přes propojovací konektor. Výstupy budičů jsou přes rezistory připojeny do hradel MOSFET tranzistorů. Tyto tvoří dva H-můstky na jejichž diagonále je připojena příslušná fáze motoru.



Obrázek 3.8 Blokové schéma výkonové části řídicí jednotky

3.2.1 Výběr spínacích tranzistorů

Pro výběru MOSFET tranzistoru s N kanálem byly stanoveny následující základní požadavky:

- Maximální přípustné napětí drain-source U_{DS} nejméně 100 V .
- Maximální proud I_D alespoň 10 A trvale.
- Hodnota odporu v sepnutém stavu $R_{DS(on)} < 0,05\ \Omega$.
- Pouzdro *TO220*.
- Přijatelná cena.

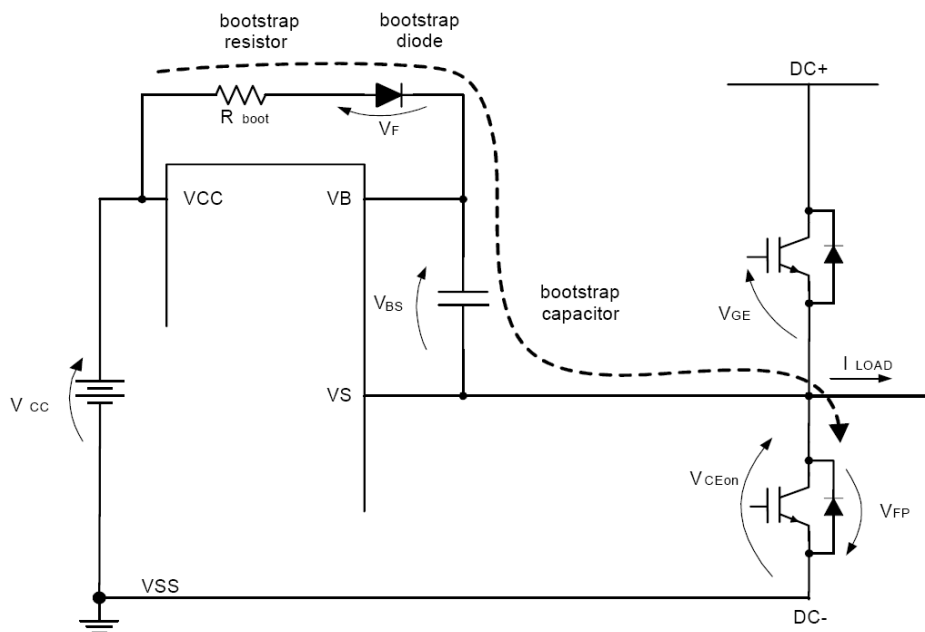
⁴ International Rectifier

Z množství zvažovaných typů byl zvolen tranzistor firmy IR, IRF3710. Tento tranzistor převyšuje stanovené požadavky, zejména hodnota odporu v sepnutém stavu $R_{DS(on)} = 23\text{ m}\Omega$ je vynikající. Jedním z důvodů pro výběr produktu firmy IR byla jeho velmi dobrá dostupnost v ČR. Cena je vyšší (36 Kč v listopadu 2006) avšak vzhledem k parametrům akceptovatelná.

3.2.2 Řízení unipolárních tranzistorů

Unipolární tranzistory MOSFET jsou řízeny elektrickým polem. Jejich hradlo je izolováno od ostatních částí polovodičové struktury a ovládací výkon je díky tomu téměř nulový. Při použití MOSFET tranzistoru jako spínacího prvku jsou důležité jeho dynamické vlastnosti. Je třeba brát v potaz poměrně významné parazitní kapacity C_{DS}, C_{GS}, C_{GD} , které jsou důsledkem struktury tranzistoru. Při spínání je pak potřeba do hradla přivést náboj (řádově desítky nC). Až ve chvíli kdy se vstupní kapacity hradla nabijí na prahové napětí $U_{GS(th)}$, se začne tranzistor otevírat. Uvědomíme-li si vztah $I = \frac{Q}{t}$, tedy že proud je náboj za čas a vzneseme-li požadavek na rychlé otevírání tranzistoru, pak je patrné, že hradlo tranzistoru nemůže být ovládáno přímo mikrokontrolérem. Výstupní piny mikrokontroléru zdaleka nejsou schopny dodat potřebný proud (i několik set mA). Musí tedy být v obvodu zařazen jakýsi mezistupeň mezi mikrokontrolérem a hradlem tranzistorů, který bude schopen dodávat krátkodobé proudové impulsy. Tímto mezistupněm je budič IR2112.

Dalším důvodem, jenž potvrzuje nutnost použití budiče, je spínání horního tranzistoru můstku. Jsou-li tranzistory zapojeny do H-můstku dle Obrázek 2.5, pak source horního tranzistoru můstku bude „plavat“ spolu se zátěží a jak bylo řečeno výše, napětí na hradle musí být o $U_{GS(th)}$ větší než napětí na source. Toho nelze dosáhnout zcela triviálně. Nejjednodušší metodou je použít zmíněný budič, který využívá zapojení *bootstrap* diody a kondenzátoru. Jak ukazuje Obrázek 3.9, kondenzátor se přes diodu nabíjí během doby, kdy je sepnut spodní tranzistor můstku. Napětí na kondenzátoru je potom využito k otevření horního tranzistoru. Podrobnější informace lze nalézt v aplikačních poznámkách IR např. [6].



Obrázek 3.9 Nabíjení *bootstrap* kondenzátoru během sepnutí spodního tranzistoru (obrázek převzat z [6])

Budiče IR2112

Jsou to vysokonapěťové budiče MOSFET a IGBT tranzistorů s kanálem typu N. Umožňují nezávislé spínání dvou tranzistorů v high a low větvi. Mohou pracovat s napětím až 600V. Zpoždění výstupu při změně stavu na vstupu obvodu je typicky 100ns. Při spínání/rozpínání tranzistorů je obvod schopen dodat/pojmout proud 200/420mA.

3.2.3 Napájení řídicí jednotky

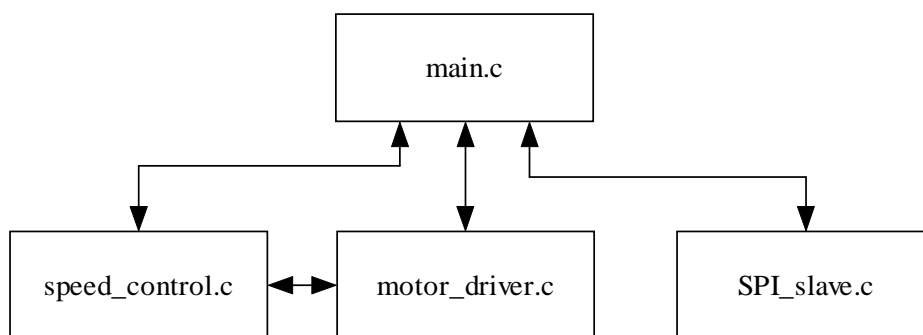
Řídicí jednotka vyžaduje ke své funkci trojici napájecích napětí. Pro logickou část je nutné stabilizované napětí 5 V, z něhož je odebíráno přibližně 50 mA. Budiče tranzistorů IR2112 jsou napájeny 5 a 12 V, proudový odběr z 12 V větve je max. 10mA. Třetí napájecí větví je napájení motoru. Toto napětí se může pohybovat v rozmezí 15–80 V a zdroj musí být proudově dimenzován dle použitého motoru. Odebíraný proud motorem by však neměl překračovat 4 A.

3.3 Software řídicí jednotky

Programové vybavení řídicí jednotky bylo vytvořeno v AVR studiu, což je integrované vývojové prostředí poskytované výrobcem mikrokontrolérů ATMEL zdarma. AVR studio je po doplnění o GCC kompilátor zdrojových kódů v jazyce C (WinAVR) plnohodnotným vývojovým prostředím, které obsahuje i simulátor, debugger a je z něho možné cílový mikrokontrolér přímo programovat. Zdrojový kód programu jsem psal v jazyce C. Použití efektivnějšího assembleru je v tomto případě prakticky vyloučené z důvodu rozsáhlosti programu a tím i potřebné doby na jeho vývoj.

3.3.1 Struktura projektu

Z důvodu přehlednosti a usnadnění orientace ve zdrojovém textu bylo nutné tento rozdělit do několika souborů.



Obrázek 3.10 Struktura souborů a jejich závislostí v projektu

Ke každému souboru .c patří příslušný hlavičkový soubor .h, který obsahuje definice proměnných a funkcí, jejichž prostřednictvím mezi sebou „komunikují“ jednotlivé .c soubory.

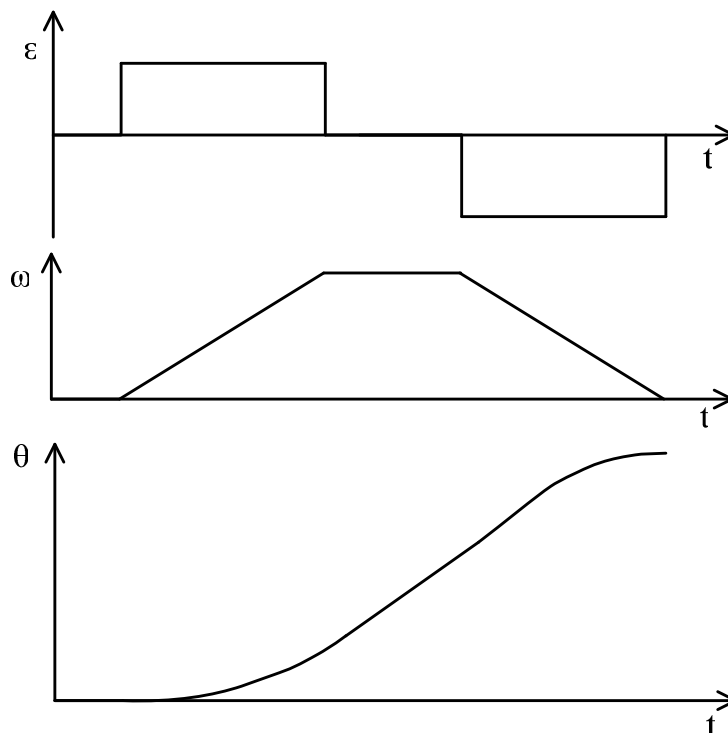
Obsah a funkce jednotlivých souborů:

- `speed_control.c`: Generování rychlostního profilu.
- `motor_driver.c`: Vyhodnocování zpětné vazby (obsluha přerušení AD převodníku), generování PWM signálu, řídicí signály pro motor
- `SPI_slave.c`: Obsluha přerušení příjmu znaku po SPI, rozpoznání významu přijatých dat, obsluha FIFO bufferu pohybových dat.

3.3.2 Lineární řízení rychlosti otáčení krokového motoru

Jak bylo uvedeno v kapitole 1.1 krokové motory mohou pracovat v otevřené smyčce řízení, proto je ale nezbytně nutné zajistit, aby při vyslání pokynu z řídicí jednotky o vykonání kroku k tomu s jistotou došlo. Motor v žádném případě nesmí ztrácet kroky, jinak by nebylo možné zachovat informaci o poloze rotoru. Z tohoto důvodu je nezbytné dimenzovat pohon tak, aby jeho zatížení nikdy nepřevýšilo točivý moment motoru.

Při dynamických změnách pohybu (rozběh, brzdění) hraje výraznou roli moment setrvačnosti rotoru a zátěže na něj připojené J a rychlost změny úhlové rychlosti rotoru $\frac{d\omega}{dt}$ – viz. vztah (2.V). Pak je patrné, že není možné, aby se rychlost otáčení rotoru měnila skokově (zrychlení by mělo tvar Diracova impulsu), ale je nutné zajistit plynulé zvyšování (snižování) rychlosti rotoru. Tak lze zaručit, že vlivem dynamických změn při pohybu nebude docházet ke ztrátám kroků.



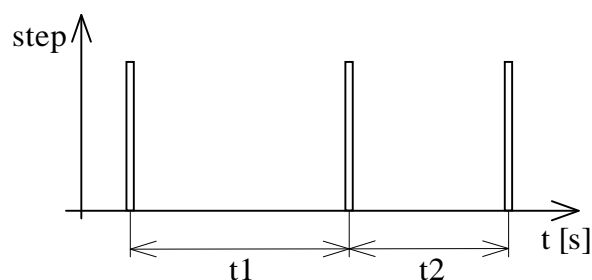
Obrázek 3.11 Příklad průběhu kinematických veličin (zrychlení, rychlost, dráha) pohonu

3.3.3 Algoritmus výpočtu časové konstanty pro lineární řízení rychlosti

V souboru *speed_control.c* je realizován algoritmus pro plynulý rozběh a zastavení motoru. Podstata tohoto výpočetního algoritmu byla publikována D. Austinem ve vědeckém magazínu „Embedded system design“ [8].

Současné mikrokontroléry umožňují provádět výpočty pro generování rychlostního profilu v reálném čase tj. již není nutné mít v paměti MCU uloženu tabulku předdefinovaných hodnot, což bylo obvyklé v minulosti, kdy výkon mikrokontrolérů nebyl pro podobné výpočty dostatečný.

Rychlost otáčení rotoru krokového motoru je nepřímo úměrná časové prodlevě mezi proudovými impulsy ve vinutí. Úkolem tohoto výpočetního algoritmu je tedy generovat časovou prodlevu mezi kroky tak, aby se rychlost otáčení motoru plynule zvyšovala či snižovala.



Obrázek 3.12 Zpoždění mezi pulsy

Prodleva mezi jednotlivými kroky je generována 16bitovým časovačem pracujícím na frekvenci f_T v módu CTC (Clear timer on compare) viz [7]. Hodnotu při níž dojde k vyvolání přerušení a resetu časovače označím c_T

$$t_l = \frac{c_{Tl}}{f_T} \quad (3.I)$$

t_l je zpoždění vygenerované časovačem při hodnotě c_T a frekvenci f_T . Úkolem je tedy určit v každém kroku novou hodnotu c_T .

$$c_n = c_0 (\sqrt{n+1} - \sqrt{n}) \quad (3.II)$$

$$c_0 = f_T \sqrt{\frac{2}{\epsilon}} \quad (3.III)$$

Vztahy (3.II) a (3.III) jsou převzaty z [8], kde lze nalézt i jejich odvození. Ve vztahu (3.II), který je nutno cyklicky počítat se však vyskytují dva výpočty odmocniny. Tato matematická operace je pro mikrokontroléry stále poměrně náročná a trvá nezanedbatelnou dobu. Z tohoto důvodu je výpočet odmocniny nahrazen aproximací pomocí Taylorova polynomu (3.IV).

$$\sqrt{1 \pm \frac{1}{n}} = 1 \pm \frac{1}{2n} - \frac{1}{8n^2} + o\left(\frac{1}{n^3}\right) \quad (3.IV)$$

Vyjádříme-li podíl $\frac{c_n}{c_{n-1}}$ podle vztahu (3.II) a dosadíme aproximaci dle (3.IV) se zanedbáním členů vyšších řádů, pak dostáváme vztah $\frac{c_n}{c_{n-1}} = \frac{4n-1}{4n+1}$, jenž lze přepsat do výsledného tvaru:

$$c_n = c_{n-1} \frac{2c_{n-1}}{4n+1} \quad (3.V)$$

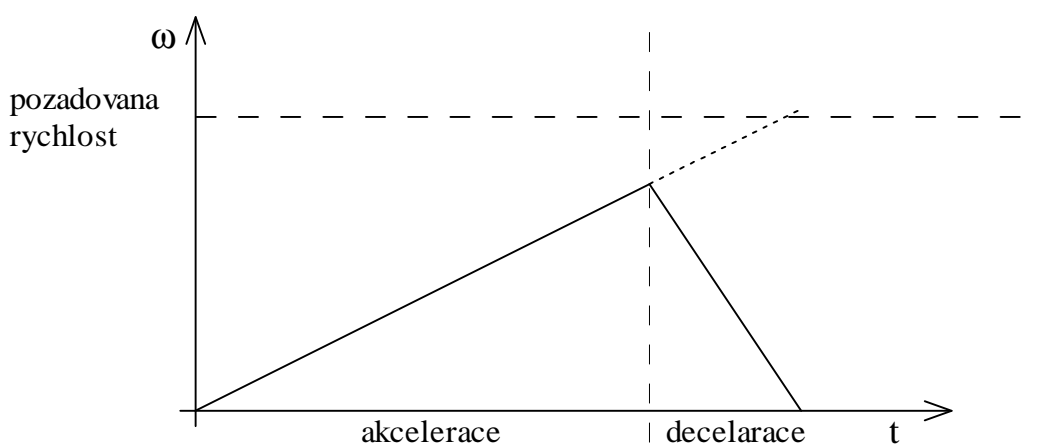
Vztah (3.V) již neobsahuje žádný typ výpočtu, díky němuž by nebylo možné tento krok cyklicky vyhodnocovat. Nejnáročnější operací je v tomto případě dělení. To je samozřejmě počítáno celočíselně a vždy je vyhodnocen zbytek po dělení pomocí operace *modulo*. Zbytek je vždy v následujícím kroku doplněn do vztahu (3.V), čímž je dosaženo zpřesnění výpočtu.

Popsaný výpočet se provádí v obsluze přerušení 16bitového časovače T1. Výsledek je uložen jako hodnota porovnávacího registru časovače T1 (OCR1A), která určuje dobu, za jakou opět toto přerušení nastane.

3.3.4 Generování rychlostního profilu

Všechny parametry požadovaného pohybu (rychlost, zrychlení, směr, počet kroků) jsou uloženy ve struktuře *SpeedData*. Jednou z položek této struktury je *Status*, proměnná, která označuje část rychlostního profilu v němž se při vykonávání pohybu jednotka právě nachází. Rozlišeny jsou stavy: akcelerace, decelerace, konst. rychlost a stop.

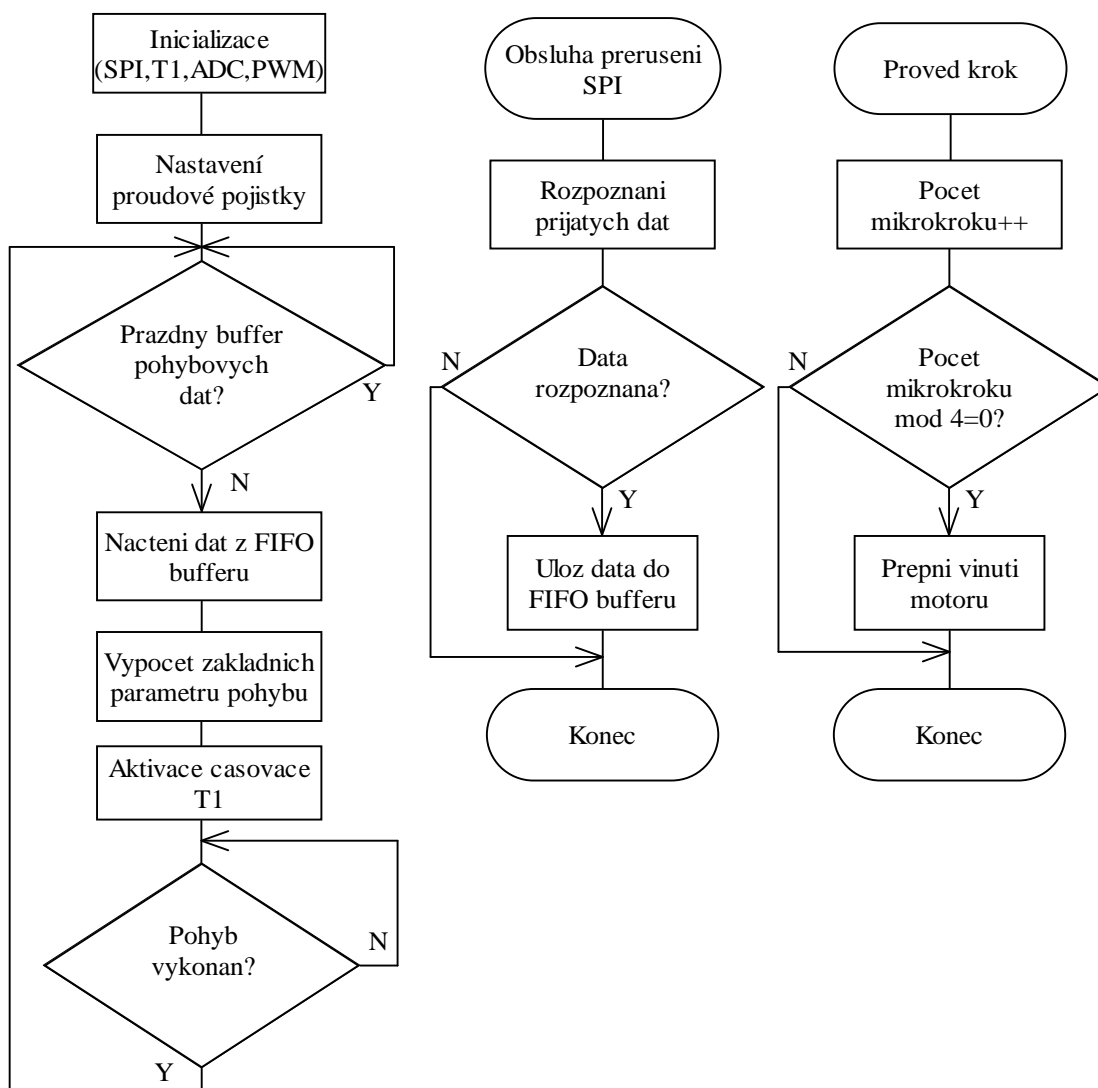
Pokud je zadání požadavků na rychlostní profil korektní, jednotka projde postupně stavy stop – akcelpace – konst. rychlost – decelerace – stop. Bylo ovšem také nutné vyřešit zadání rychlostního profilu takové, kdy je požadovaný počet kroků (tedy úhlová dráha) příliš nízký k dosažení vyžadované rychlosti při zadaném zrychlení. V takovém případě vygeneruje jednotka rychlostní profil tak, aby při zachování požadované akcelpace a decelerace byl vykonán žádaný počet kroků a na zadanou rychlost se nebere ohled. Rychlostní profil se pak skládá pouze ze stavů stop – akcelpace – decelerace – stop.



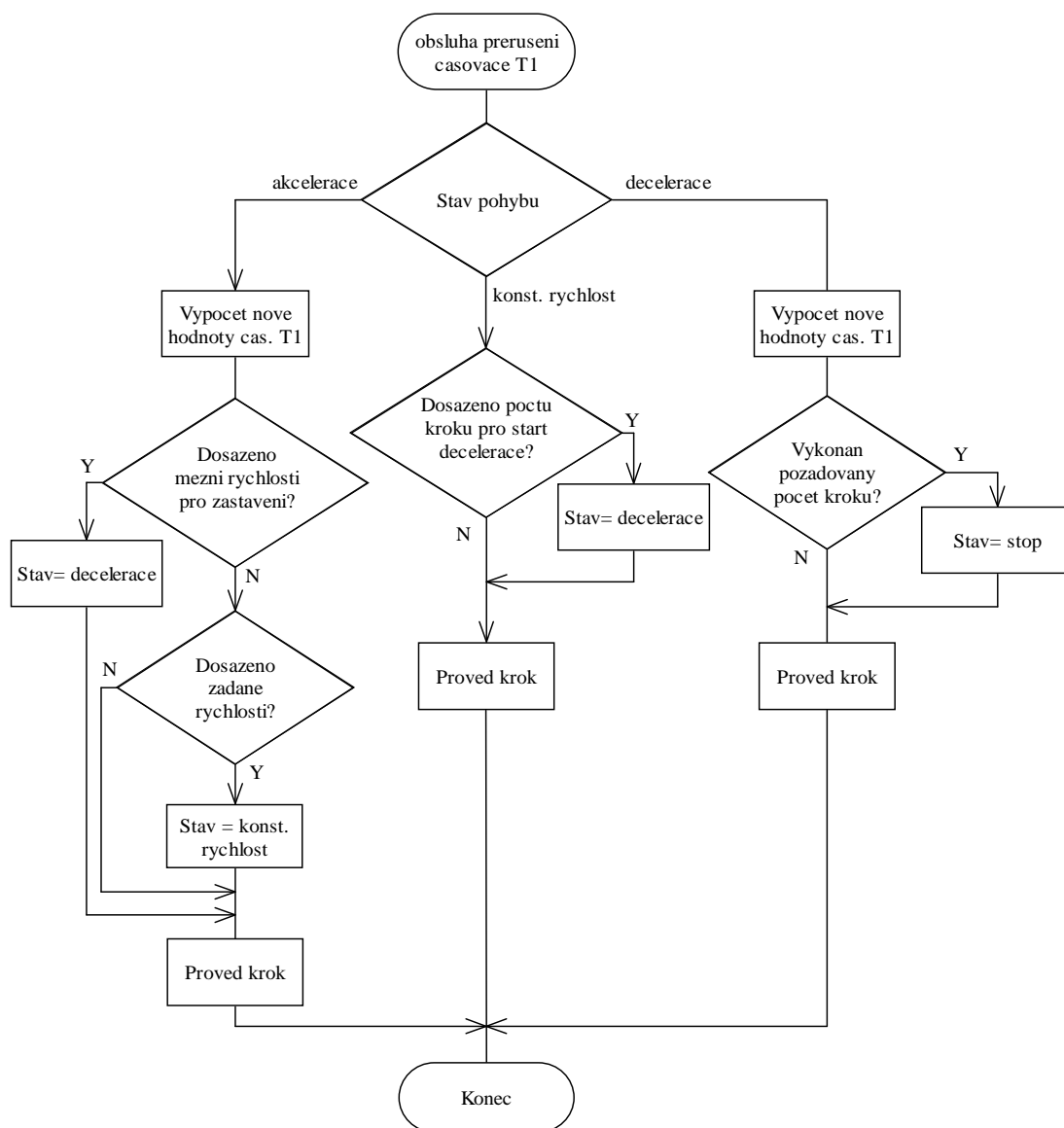
Obrázek 3.13 Příklad rychlostního profilu, kdy není dosaženo požadované rychlosti

3.3.5 Vývojové diagramy programového vybavení řídicí jednotky

Z vývojových diagramů si lze udělat základní představu, jak SW této jednotky pracuje. Blíže lze SW pochopit nahlédnutím do zdrojových textů, které jsou součástí příloženého CD. Zdrojový text je samozřejmě okomentován.



Obrázek 3.14 Základní vývojové programy SW vybavení



Obrázek 3.15 Vývojový diagram obsluhy přerušení časovače T1

4 Řídící jednotka polohovacího systému

Tato jednotka představuje hlavní řídicí prvek celého systému, je nadřazena jednotkám motorů v jednotlivých osách a vytváří HMI⁵. V dalším textu je řídicí jednotka polohovacího systému nazývána též Master jednotkou.

Hlavní požadavky kladené na jednotku:

- a) Příjem povelů od uživatele.
- b) Výpočet parametrů pohybu k dosažení žádané polohy a předávání těchto informací jednotkám motorů.
- c) Zpracování signálu snímačů koncové polohy.
- d) Uchování informace o reálném čase.
- e) Vstup dat z klávesnice, výpis informací na LCD displej.
- f) Snadná programovatelnost a možnost budoucího rozšíření.

a) Příjem povelů od uživatele má být umožněn dvěma způsoby. Buď ovládáním z PC pomocí příkazů specifikace SCPI (tzv. remote režim), nebo vkládáním příkazů pomocí 16tlačítkové klávesnice (autonomní režim). Pro remote mód mají být hardwarově realizovány rozhraní RS232, USB a Bluetooth, přičemž software má být napsán pro první dvě.

b) Parametry pohybu mají být vyhodnoceny z přijatých příkazů, vhodně upraveny a přes SPI rozhraní předány jednotkám motorů v příslušných osách.

c) Signály snímačů koncové polohy je nutné upravit tak, aby odpovídaly napětovým úrovním TTL logiky, sekvenčně je vyhodnocovat a žádaným způsobem tím omezit pracovní prostor polohovacího systému. Zvolené koncové snímače zároveň tvoří referenční bod systému, k němuž jsou vztaženy veškeré požadavky polohování.

d) Pro plánované využití polohovacího systému k měření v magnetických polích je žádoucí, aby řídicí jednotka umožnila uchování reálného času (i při odpojení napájecího napětí) z důvodu pozdější identifikace naměřených dat.

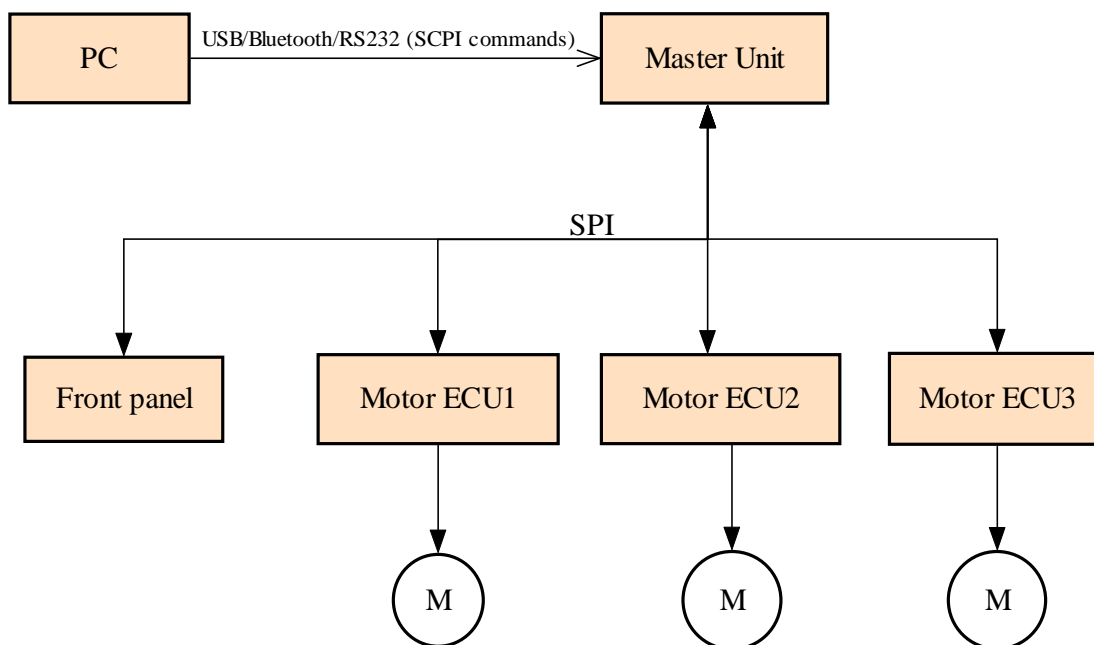
⁵ HMI – Human-Machine Interface, termín používaný pro označení rozhraní člověk - stroj

e) K jednotce má být připojen čelní panel obsahující 16tlačítkovou numerickou klávesnici a k výpisu informací víceřádkový LCD displej.

f) Kvůli snadnému programování a ladění bylo požadováno osadit řídicí jednotku mikrokontrolérem s rozhraním JTAG. Z důvodu plánovaného rozšiřování ovládacího SW má mikrokontrolér disponovat dostatkem programové paměti.

4.1 Topologie celého řídicího systému

Obrázek 4.1 zobrazuje navrženou topologii celého systému řízení. Navržená struktura se skládá ze třech vrstev. Nejnižší vrstvou jsou akční členy – krokové motory, které zajišťují pohon příslušných os polohovací soustavy. Každý z motorů je ovládán samostatnou řídicí jednotkou, již je nadřazena řídicí jednotka polohovacího systému. Tato, jak již bylo řečeno, přijímá příkazy od uživatele buď prostřednictvím vzdáleného PC nebo čelního panelu.



Obrázek 4.1 Navržená topologie

4.2 Návrh elektroniky jednotky

Prvním krokem při návrhu této jednotky byl opět, stejně jako v případě řídicích jednotek krokových motorů, výběr vhodného mikrokontroléru. Jedná se o zásadní část návrhu, od níž se později odvíjí možnosti a budoucí rozšiřitelnost realizovaného řídicího systému.

4.2.1 Výběr mikrokontroléru

Při výběru vhodného mikrokontroléru byly zvažovány pouze produkty firmy ATMEL a to z důvodu zakoupení programátoru JTAGICE mkII, který umožňuje programování 8bitových mikrokontrolérů řady AVR zmiňovaného výrobce.

Hlavní požadavky na mikrokontrolér:

- Dostatek I/O portů pro připojení klávesnice, koncových snímačů, ovládání LCD displeje a pomocných komunikačních signálů jednotek motorů.
- Minimálně dvojice UART kanálů.
- Rozhraní SPI, JTAG.
- Dostatek programové paměti pro možnost výrazného budoucího SW rozšíření.

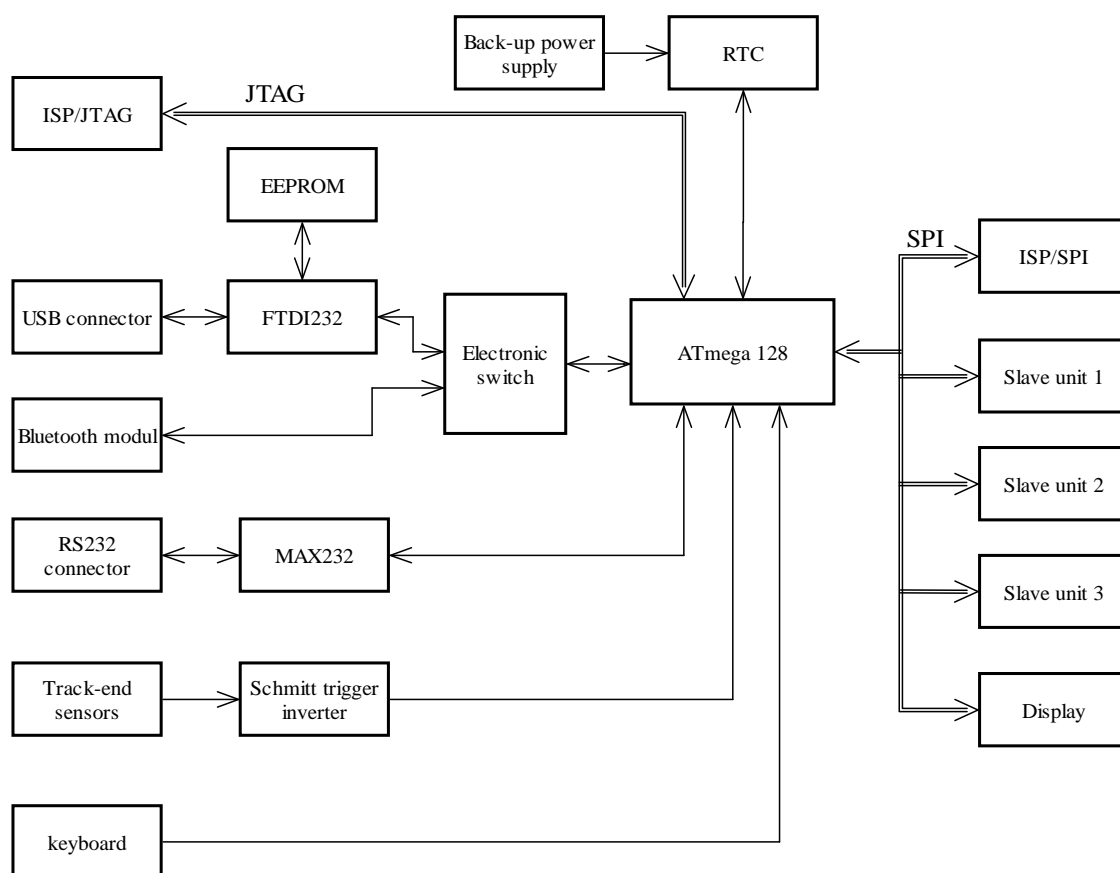
Po úvaze byl zvolen mikrokontrolér ATmega128, který velmi dobře splňuje výše uvedené požadavky a zejména velikost programové paměti vyhovuje vysokým nárokům na pozdější rozšíření SW vybavení. Cena mikrokontroléru byla (začátek roku 2007) cca 150 Kč.

Hlavní parametry mikrokontroléru ATmega128:

- 128K Byte paměti FLASH pro program, 4 Kbyte SRAM,
- 2×16 bit časovač, 2×8 bit časovač,
- programování v cílové aplikaci (ISP) přes SPI nebo JTAG rozhraní,
- pracovní napětí 4,5–5,5V, pouzdro TQFP64,
- struktura mikrokontroléru optimalizována pro programování v jazyce C.

4.2.2 Blokové schéma řídicí jednotky

Na následujícím blokovém schématu lze dobře znázornit strukturu navrhované řídicí jednotky. Konkrétní schéma zapojení všech elektronických prvků je součástí přílohy [B]. Celé zařízení je napájeno stabilizovaným napětím 5 V a odběr proudu nepřesahuje 200 mA.



Obrázek 4.2 Blokové schéma řídicí jednotky polohovacího systému

Popis jednotlivých částí blokového schématu

Mikrokontrolér

Srdcem jednotky je mikrokontrolér ATmega128, který zpracovává veškeré vstupní signály a řídí všechny výstupy. Celá činnost jednotky je tak na mikrokontroléru závislá, zejména pak na kvalitě a možnostech jeho software.

Rozhraní pro komunikaci s PC

Ke komunikaci s PC jsou připravena tři různá rozhraní RS232, USB a Bluetooth. Protože mikrokontrolér ATmega128 disponuje pouze dvěma UART kanály, a nepředpokládá se, že by bylo nutné, aby jednotka komunikovala přes USB i Bluetooth zároveň, použil jsem zapojení elektronického přepínače s obvodem 4066. Tedy USB a Bluetooth jsou připojeny k jednomu UART kanálu a aktivní může být v daný okamžik pouze jedno z těchto rozhraní. Druhý UART kanál mikrokontroléru je použit pro komunikaci přes RS232.

V případě USB rozhraní jsem použil obvod FTDI232 jako převodník USB/UART. Po nainstalování ovladačů dodávaných výrobcem FTDI obvodů se v PC objeví nový virtuální sériový port. Veškerý vývoj SW, jak na straně PC, tak na straně mikrokontroléru, je potom výrazně zjednodušen, protože není nutné znát poměrně složitou a obsáhlou specifikaci USB. SW je tak vyvíjen stejně jako při komunikaci přes skutečný sériový port. Obvod FTDI byl doplněn pamětí 93C46, což je EEPROM paměť velikosti 128x8 byte. Pomocí SW dodávaného výrobcem FTDI lze do paměti uložit konfigurační údaje např. sériové číslo zařízení, nastavení napájení a též jméno pod nímž se zařízení hlásí po připojení k PC. Dále jsem k FTDI obvodu připojil LED diody, které slouží k indikaci právě probíhajícího přenosu dat. Při vývoji SW pak LED diody usnadňují ladění.

Pro realizaci komunikace přes RS232 bylo z HW hlediska pouze nutné přizpůsobení napětíových úrovní obvodem MAX232, což je velmi známý převodník RS232/TTL. K MAX232 byly opět připojeny indikační LED diody.

Zvolil jsem nejjednodušší třívodičové zapojení sériové linky. Nepředpokládá se totiž ani velká vzdálenost PC od připojené řídicí jednotky ani provoz v prostředí s vysokým rušením. Během testování nedošlo k jedinému případu ztráty posílaných dat, proto lze tvrdit, že toto jednoduché zapojení je pro dané použití dostačující.

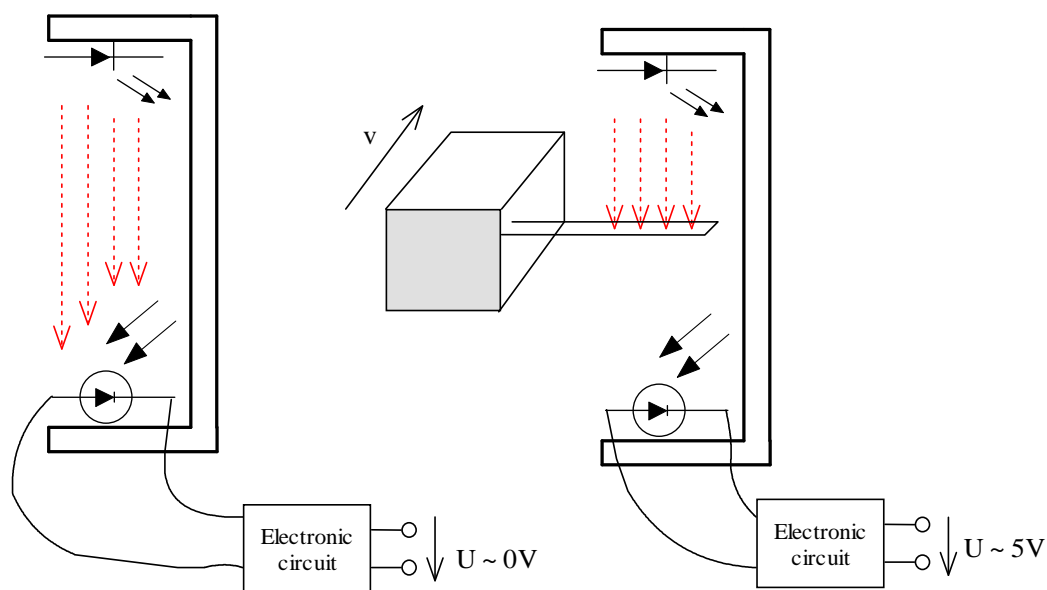
K realizaci Bluetooth komunikace byl použit modul AMB2300 výrobce AMBER wireless. Modul obsahuje rozhraní UART, proto je možné ho přímo připojit k mikrokontroléru. Po doplnění SW vybavení bude jednotka umožňovat bezdrátovou komunikaci se vzdáleným zařízením. Realizace tohoto SW však přesahuje rámec této DP a svým rozsahem by mohla být samostatným projektem či DP.

Obvod reálného času

Jako RTC je použit obvod DS1302 výrobce Dallas semiconductor, který umožňuje uchování času, data, dnu v týdnu a bere úvahu dokonce přestupné roky. Komunikace mikrokontroléru s DS1302 je synchronní sériová po 3 vodičích. Jedinou vnější součástí potřebnou pro funkci obvodu je krystal pracující na frekvenci 32,768 kHz. Pro udržení informace o čase i v případě absence napájecího napětí je jako záložní zdroj energie připojena standardní 3V lithiová baterie.

Snímače koncové polohy

Kvůli finanční úspoře nejsou použity komerční bezdotykové koncové snímače, nýbrž velmi jednoduché snímače vlastní konstrukce. Ty pracují na fotoelektrickém principu způsobem často označovaným jako „optická závora“.



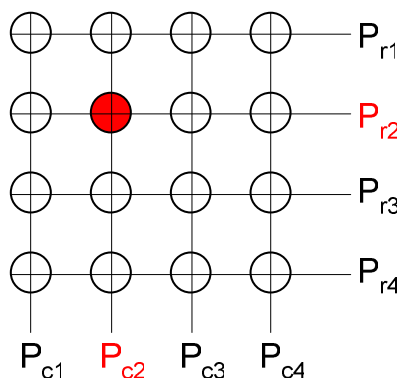
Obrázek 4.3 Ilustrace funkce optické závory

Na každém rameni polohovacího systému je v drážce připevněn kovový plíšek sloužící jako clonka. Optické závory jsou umístěny tak, aby clonka přerušila světelný paprsek, pokud se dané rameno blíží své krajní poloze. Protože je výstupní napětí snímače v klidovém stavu rovno přibližně 0,9V, zařadil jsem do obvodu řídicí jednotky Schmittův invertor s hysterezí 74HCT14. Tento obvod zajišťuje korekci napěťových úrovní, protože dle specifikace TTL logiky je signál ze snímače v klidovém stavu v zakázaném pásmu.

Pokud by tento obvod nebyl použit, mohlo by se v krajním případě stát, že by signál ze snímačů byl mikrokontrolérem chybně vyhodnocen

Klávesnice

V autonomním režimu ovládání polohovacího systému je prostředkem pro zadávání příkazů běžná maticová klávesnice obsahující 4×4 tlačítek. Pro připojení klávesnice je vyhrazeno osm I/O pinů mikrokontroléru.



Obrázek 4.4 Ilustrace zapojení maticové klávesnice

Pomocí P_{r1} – P_{r4} jsou označeny řádky, pomocí P_{c1} – P_{c4} sloupce matice. Vyhodnocování stisknutého tlačítka pracuje na jednoduchém principu: piny mikrokontroléru připojené k P_{r1} – P_{r4} jsou v první okamžik nastaveny jako výstupní a P_{c1} – P_{c4} jako vstupní. Stav pinů P_{c1} – P_{c4} je vyhodnocen a tím získáme informaci o tom, zda je stisknuto nějaké tlačítko popř. ve kterém sloupci. V druhém kroku je situace obdobná. Piny P_{r1} – P_{r4} nastaveny jako vstupní, P_{c1} – P_{c4} jako výstupní. Získáváme informaci o případně stisknutém tlačítku v některé z řádek.

Dle stavu ve sloupcích a řádcích je již jednoduché identifikovat, které tlačítko bylo stisknuto.

SPI rozhraní

Pomocí SPI rozhraní jsou připojeny podřízené jednotky krokových motorů, též je přes toto rozhraní možné mikrokontrolér programovat. Slouží také ke komunikaci mikrokontroléru s displejem – k zasílání konfiguračních údajů a znaků, které chceme na displeji zobrazit.

Displej

Jako zobrazovací prvek byl zvolen displej EADOGM163B-A výrobce Electronic assembly. Jedná se o třířádkový LCD display, přičemž na každé řádce je možno zobrazit až 16 znaků. Řadičem displeje je obvod ST7036. Velkou výhodou je, že displej, mimo paralelního zápisu dat, umožňuje i komunikaci přes SPI rozhraní. Tím jsou zaručeny minimální nároky na počet obsazených pinů mikrokontroléru (mimo SPI pouze další dva).

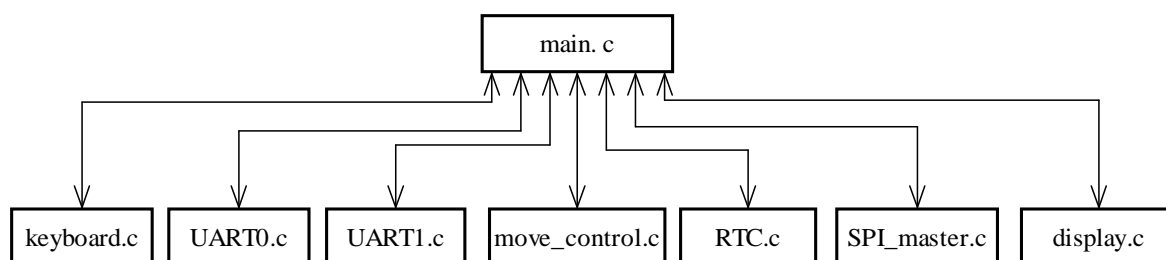
4.3 Software řídicí jednotky

Pro tvorbu tohoto SW platí stejné obecné informace jako již uvedené v kapitole 3.3. V porovnání se SW řídicí jednotky krokových motorů je tento výrazně rozsáhlejší a náročnější na programovou i operační paměť mikrokontroléru, avšak algoritmická složitost je zhruba na stejné úrovni.

4.3.1 Struktura projektu

Programový kód bylo nutné rozdělit do několika souborů kvůli zvýšení přehlednosti a též za účelem snadnějšího ladění. Nevýhodou tohoto způsobu rozdělení zdrojového textu do více souborů je nutnost definovat „styčné body“ v podobě funkcí a globálních proměnných, kterými jsou jednotlivé části kódu propojeny. Programový kód délky několika tisíc řádků však do jednoho hlavního souboru zkrátka umístit nelze.

Ke každému souboru .c přísluší hlavičkový soubor .h s definicí příslušných proměnných a funkcí.



Obrázek 4.5 Rozdělení projektu do souborů

4.3.2 Obsah jednotlivých zdrojových souborů

Main.c

Základní zdrojový soubor z něhož jsou volány funkce definované v ostatních souborech. Jsou zde vyhodnocovány SW příznaky, které se nastavují i v dalších souborech a označují vykonání určité části kódu jako reakci na nějakou událost např. příznak zpracování dat přijatých po USB či RS232, příznak rozpoznání vstupu dat z klávesnice atd. Podstatnou součástí tohoto souboru je rozpoznávání a interpretace přijatých příkazů z PC.

Hlavní část kódu je prováděna v nekonečné programové smyčce, z níž se „odskakuje“, pokud nastane žádost o přerušení nebo při vykonávání funkcí obsažených v dalších souborech.

UART0.c, UART1.c

V těchto souborech je kód k obsluze UART kanálů. Kód obou souborů je dosti podobný, UART0.c zpracovává data přijatá (a odesílaná) po RS232, zatímco UART1.c data přenášena po USB. V obou souborech jsou realizovány přijímací a odesílací FIFO buffery, do kterých jsou ukládány přijaté a odesílané znaky.

Přijímané znaky jsou ukládány do bufferu v obsluze přerušení vykonané při dokončení příjmu znaku (*USART_RX_vect*). Odesílání znaku z odesílacího bufferu je vykonáno v obsluze přerušení prázdného odesílacího registru (*USART_UDRE_vect*).

Dále jsou zde definovány funkce pro práci s přijatými daty – zejména fce *Get_str()*, která vrátí požadovaný počet znaků z přijímacího bufferu a funkce pro práci se samotným přijímacím bufferem – jeho vymazání, oznámení, že je buffer plný atd.

SPI_master.c

Tento soubor obsahuje funkce pro odesílání a příjem dat po SPI. V obsluze přerušení dokončení SPI vysílání (*SPI_STC_vect*) je nastavován příznak *SPI_free*, který označuje, že SPI je volné pro další data. Zápis dat do odesílacího registru *SPDR* je možný pouze tehdy, pokud je nastaven *SPI_free*. Tím je zaručeno, že příslušná data budou skutečně odeslána.

RTC.c

Zde jsou definovány funkce *RTC_write()* a *RTC_read()*, které realizují zápis a čtení dat do/z RTC obvodu DS1302.

move_control.c

Úkolem programového kódu v tomto souboru je výpočet parametrů pohybu, který má být vykonán. Uživatelem zadané parametry pohybu jsou interpretovány tak, aby bylo možné je předat podřízeným řídicím jednotkám motorů. Například je zde vypočten počet kroků potřebný k dosažení žádané polohy, je nastavena požadovaná rychlost atd.

keyboard.c

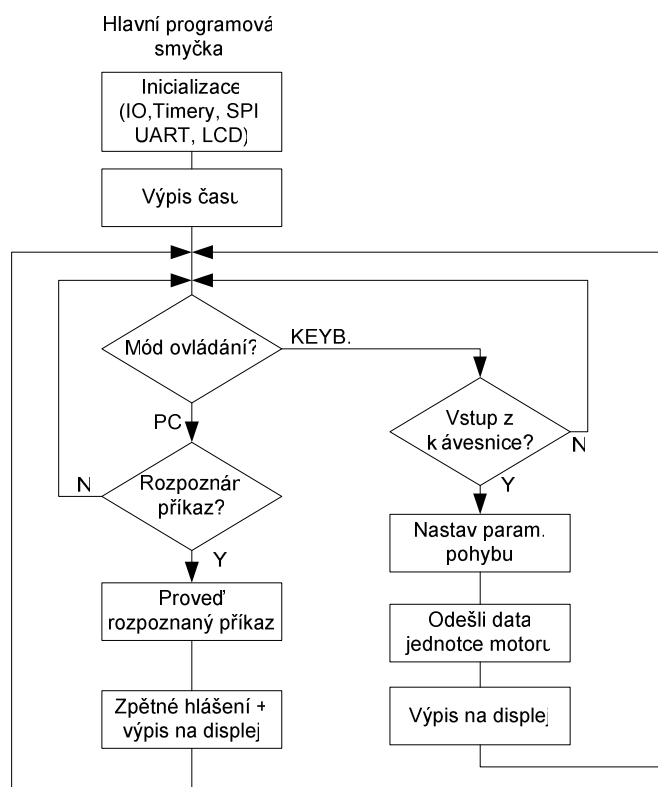
Tento soubor je určen pro práci s klávesnicí. Veškerý kód vyhodnocující stav na klávesnici je umístěn v obsluze přerušení časovače T3. Četnost jeho vykonávání je cca 4x za sekundu. Tato četnost byla nastavena experimentálně a z možných variant nastavení časovače vyhovovala nejlépe.

display.c

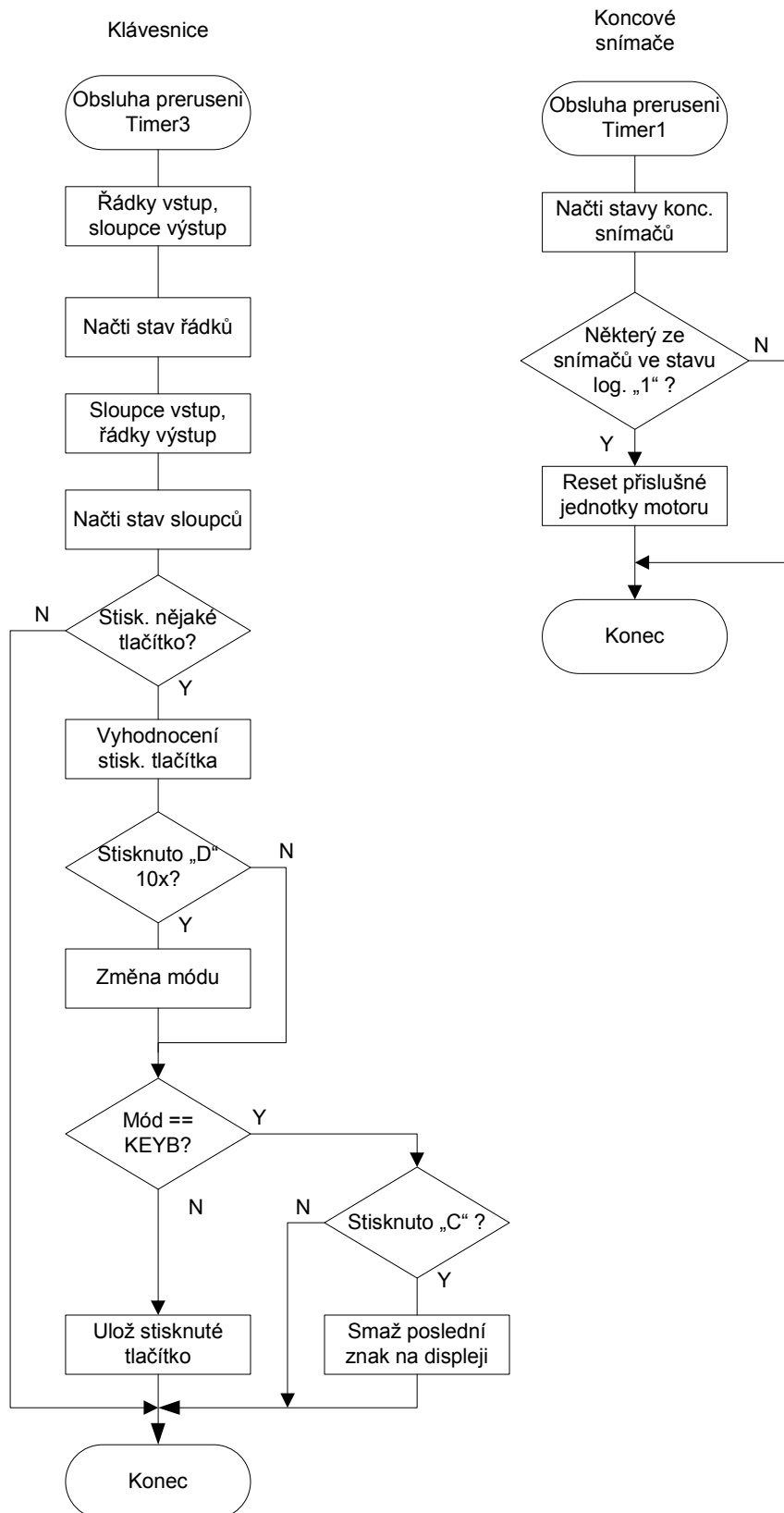
Tento soubor obsahuje kompletní sadu funkcí pro práci s displejem. Nejdůležitější z nich jsou `lcd_putc()` a `lcd_puts()`, přičemž první z nich vypíše na display předaný znak a druhá celý předaný řetězec. Dále jsou tu funkce pro smazání displeje, přechod kurzoru mezi řádky, návrat kurzoru na počátek prvního řádku atd.

4.4 Vývojové diagramy SW

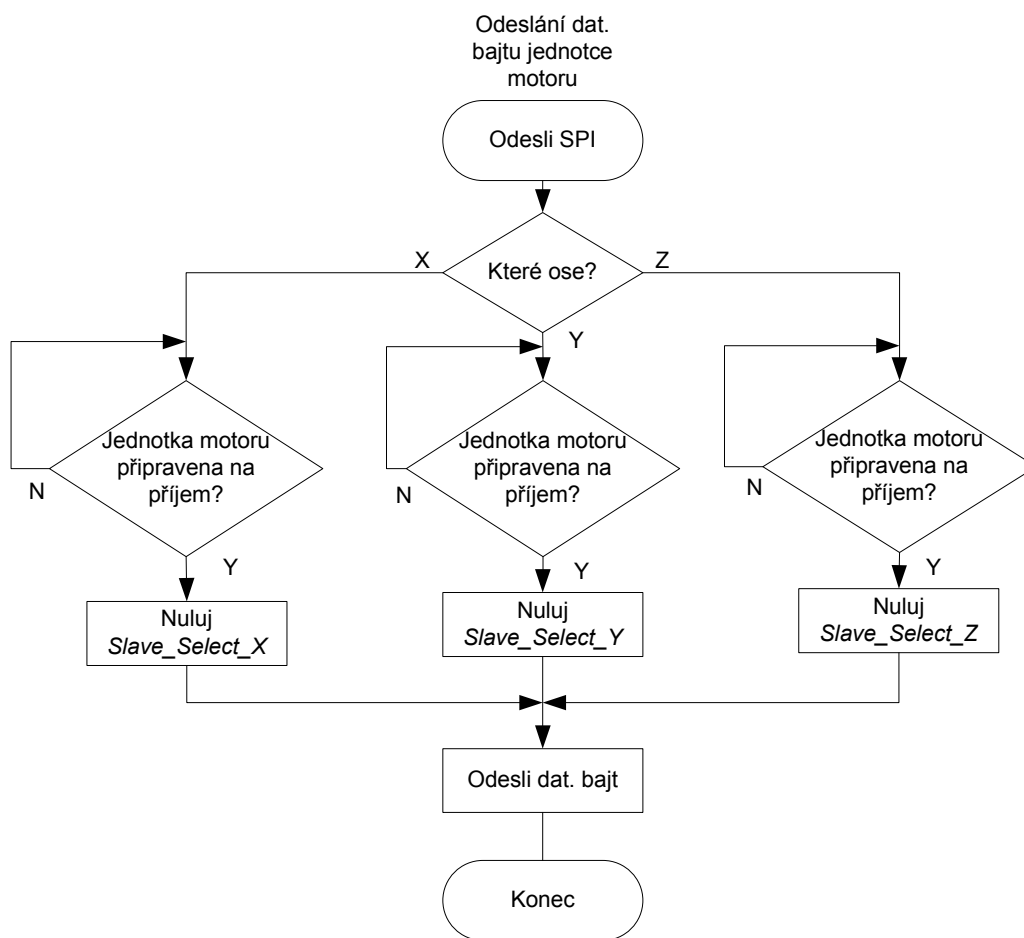
Tato kapitola obsahuje vývojové diagramy SW Master jednotky. Z důvodu rozsáhlosti jsou diagramy co možná nejvíce zestručněny. K úplnému pochopení složitějších pasáží SW lze doporučit nahlédnutí do komentovaného zdrojového kódu, který je součástí příloženého CD.



Obrázek 4.6 Vývojový diagram hlavní programové smyčky



Obrázek 4.7 Vývojové diagramy obsluhy klávesnice a koncových snímačů



Obrázek 4.8 Vývojový diagram odesílání dat po SPI

4.5 Příkazy předávané uživatelem

Dle požadavků v zadání měl být vytvořen soubor příkazů, kterými bude moci uživatel ze vzdáleného PC ovládat celý polohovací systém. Příkazy mají vycházet ze standardu SCPI⁶.

4.5.1 SCPI standard

Standard SCPI je specifikace množiny příkazů a pravidel pro ovládání číslicového měřicího přístroje nadřazeným systémem (nejčastěji PC) nezávisle na použitém protokolu přenosu dat. Přínosem SCPI je sjednocení dříve velmi rozdílných přístupů výrobců ke vzdálenému ovládání měřicího přístroje. Bylo běžné, že každý výrobce definoval svůj vlastní příkazový systém. Zavedením SCPI dochází z pohledu programátora aplikace pro komunikaci s různými měřicími přístroji ke značnému zjednodušení.

⁶ Standard Commands for Programmable Instrumentation

Příkazy jsou uspořádány do stromové struktury.

Např.: MEASure

:VOLTage

:DC? {<range>|MIN|MAX|DEF},{<res>|MIN|MAX|REF}

:AC?

Příklad příkazu: *MEAS:VOLT:DC? 1 0.001;*

Více o SCPI a číslicových měřicích systémech viz [9].

Poznámky k syntaxi:

Dvojtečka (:) slouží jako oddělovač příkazů na dvou po sobě jdoucích úrovních příkazového stromu. Středník (;) odděluje příkazy stejné úrovně. Příkazy s otazníkem (?) mají význam dotazu a je vždy očekávána odpověď přístroje.

Pro mou DP je nejdůležitější dodržet pravidla tvorby SCPI příkazů a na jejím základě stanovit nové příkazy přizpůsobené pro ovládání polohovacího systému.

4.5.2 Soubor realizovaných příkazů a jejich popis

Některé použité příkazy jsou přímo definovány v SCPI množině příkazů, jiné pouze vycházejí z její syntaxe.

***RST** – Provádí reset celého systému.

***IDN?** – Žádá identifikační číslo, slouží zejména k ověření navázání spojení.

POS

:X, Y, Z

:REL, ABS, REF;

Provádí posuv ve vybrané ose. Zadané parametry jsou chápány jako absolutní (vzhledem k bodu [0,0], který je definován umístěním koncových snímačů), nebo relativní vzhledem k poslední poloze. Parametr REF označuje požadavek na najetí do referenční polohy.

POS

:X?, Y?, Z?;

Žádost o informaci o poloze v dané ose⁷.

⁷ Informace o poloze je udržována pouze „softwarově“. Snímače skutečné polohy nejsou z důvodu ceny použity.

LCD

:CLR;

Požadavek na vymazání displeje.

SWT

:COM, USB;

Informuje o použitém komunikačním rozhraní resp. jeho změně.

INTRP

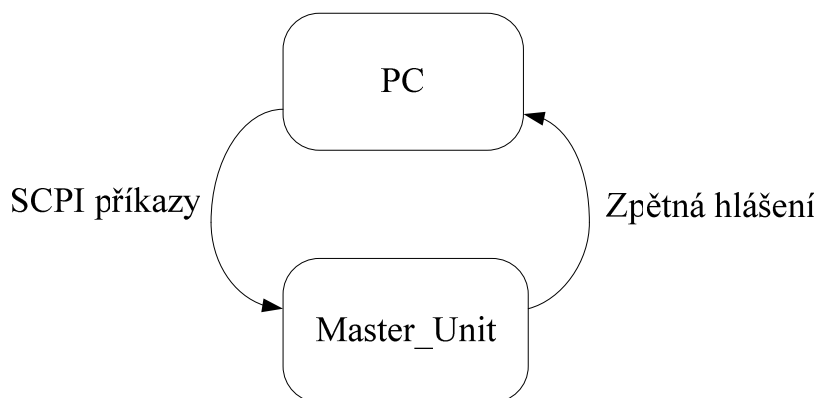
:USR, LIN;

Příkaz jednotce, aby změnila způsob výpočtu parametrů pohybu. V režimu LIN se parametry počítají tak, aby byl realizován pohyb po přímce.

Soubor příkazů není zatím příliš rozsáhlý, jsou definovány jen základní příkazy pro polohování a elementární ovládání. Pro plné využití polohovacího systému jakožto měřicího pracoviště bude podle potřeby zřejmě nutné další příkazy dodefinovat. Struktura software s tím ale počítá a rozšíření zmíněným způsobem bude velice snadné.

4.6 Předávání zpráv v rámci celého systému

Předávání zpráv v celém systému lze rozdělit do dvou vrstev. Všechny jednotky jsou propojeny viz Obrázek 3.1. Mezi vzdáleným PC a řídicí jednotkou polohovacího systému probíhá předávání SCPI příkazů a po jejich rozpoznání Master jednotka opačným směrem předává zpětná hlášení. Tato hlášení jsou určena k informování uživatele o úspěšném předání příkazu.



Obrázek 4.9 Zasílání zpráv mezi PC a Master jednotkou

Další úrovní komunikace je předávání zpráv mezi jednotkou Master a jednotkami krokových motorů. Tato komunikace již pro uživatele nemusí být transparentní a nevychází z nějakého obecně používaného standardu.

Zpráva zasílaná Master jednotkou obsahuje vždy následující parametry požadovaného pohybu: *zrychlení*, *zpomalení*, *rychlost*, *počet kroků*, *směr*. Každý údaj se skládá ze tří bajtů, mimo údaje *směr*, pro který postačují dva bajty.

První zasláný bajt vždy obsahuje písmeno označující údaj, jenž bude bezprostředně následovat: a = zrychlení, d = zpomalení, v = rychlost, s = počet kroků, w = směr. Podle těchto písmen jednotka motoru snadno rozliší právě zasílaný údaj. Poté následuje horní a dolní bajt číselné hodnoty předávaného parametru. Teoreticky by bylo možné dopředné označování pohybových parametrů písmeny vypustit, za předpokladu, že údaje jsou Master jednotkou vždy zasílány ve stejném pořadí. Domnívám se však, že z pohledu možné budoucí rozšiřitelnosti, snadnější čitelnosti kódu a pochopení všech náležitostí bylo vhodné údaje zasílané v rámci jedné zprávy tímto způsobem uvozovat, byť je tím v každé zprávě zasláno pět bajtů navíc. Každá zpráva je ukončena znakem #.

Příklad zasílané zprávy:⁸

a	1	244	d	0	200	v	3	232	s	7	208	w	L	#
---	---	-----	---	---	-----	---	---	-----	---	---	-----	---	---	---

Význam: zrychlení = 500, zpomalení = 200, rychlost = 1000, počet kroků = 2000, směr = vlevo.

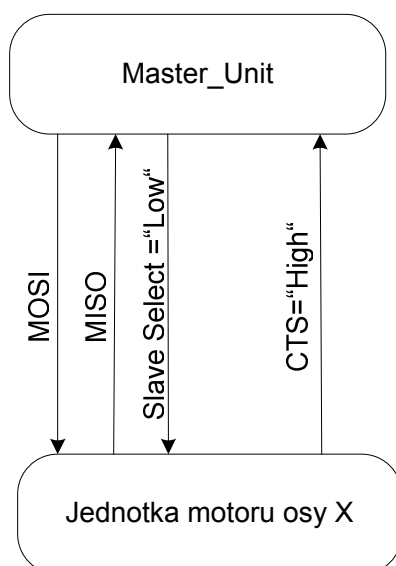
Rychlost je dána v jednotkách *krok / s*, zrychlení a zpomalení pak v *krok / s²*. Po přijetí ukončovacího znaku # jednotka motoru přijatou zprávu vyhodnotí (kontroluje maximální přípustné hodnoty) a poté ovládá motor dle zadání.

HW implementace SPI kanálu procesorů ATmega [10] nedovoluje při příjmu dat uchovat ve vstupních bufferech více než jeden bajt. Tedy v případě, že vstupní buffery obsahují ještě nepřečtený datový bajt a pokusili bychom se o zaslání dalších dat, pak nepřečtená data ze vstupního bufferu budou ztracena. Toto v praxi znamená, že při přenášení zpráv po SPI by bylo nutné všechny zprávy Master jednotce potvrzovat. Tato možnost se však zdála být poměrně složitou. Protože jsem při návrhu hardware pro komunikaci vyhradil mimo SPI ještě další tři rezervní piny – opět s ohledem na budoucí rozšíření, ukázalo se jako vhodné řešení jeden z rezervních pinů využít pro indikaci stavu, kdy je jednotka motoru zaneprázdněna resp. kdy je schopna přijímat data. Tento pin je pak v kódu nazýván *CTS – Clear to send*.

Zprávy zasílané Master jednotkou nejsou tedy jednotkami motorů potvrzovány, ale Master jednotka je vysílá pouze ve chvíli, kdy je příslušná jednotka motoru připravena data přijmout.

Jednotka motoru, které jsou data určena, je pochopitelně vybrána pomocí signálu *Slave select*. Viz kapitola o SPI 3.1.2.

⁸ Číselné hodnoty jsou v dekadickém tvaru



Obrázek 4.10 Příklad komunikace Master jednotky s jednotkou motoru

Obrázek 4.10 ukazuje signály při komunikaci mezi Master jednotkou a jednotkou motoru osy X polohovacího systému. Master jednotka stavem signálu *Slave select* log. „0“ vybírá jednotku s níž chce komunikovat. Pokud její signál CTS je v úrovni log. „1“ (jednotka motoru připravena přijímat data), pak Master odesílá jednotlivé datové bajty zprávy.

4.7 Desky plošných spojů

Poměrně pracnou a zejména časově náročnou částí této DP byl návrh desek plošných spojů. Jako návrhový systém jsem zvolil OrCAD 10 a to zejména na doporučení vedoucího práce, dále z důvodu snadno dostupné výukové literatury např. [11] a též díky možnosti využít knihovny součástek dříve vytvořené pracovníky Katedry elektrotechniky TUL. Všechny DPS (mimo čelního panelu s displejem) jsou dvouvrstvé s nepájivou maskou. DPS byly vyrobeny na zakázku v PCB-Lab na TUL.

Fotografie výsledných plošných spojů jsou součástí přílohy [C].

5 Ovládací SW pro PC

Závěrečnou úlohou této DP bylo naprogramovat aplikaci pro PC pod Windows. Tato má umožnit uživateli ovládání polohovacího systému formou zasílání příkazů dle standardu SCPI.

5.1 Tvorba a popis aplikace

K vývoji jsem zvolil prostředí Borland C++ Builder. Důvodem pro tuto volbu byla zejména možnost použití knihovny VISA⁹, což je soubor funkcí sloužících pro přenos dat v automatizovaných systémech měření. V mém případě lze VISA funkcí s výhodou využít pro komunikaci s připojenou řídicí jednotkou a to jak přes RS232, tak přes USB.

5.1.1 Zdrojový kód

Zdrojový kód aplikace je poměrně jednoduchý – jeho hlavním úkolem je uživatelem vybraný povel se zvolenými parametry převést do vhodného textového řetězce tak, aby výsledný příkaz splňoval SCPI standard. Příkaz je pak přes uživatelem zvolené komunikační rozhraní přenesen do řídicí jednotky. K tomu jsem použil funkci *viQueryf()* z knihovny VISA. Tato funkce odešle předaná data a poté zvolenou dobu (timeout) čeká na odpověď. Pokud odpověď v této době přijde je zapsána do připraveného textového pole.

Výslednou aplikaci bude zřejmě nutné výrazně rozšířit dle konkrétních požadavků při měření, které ale při její tvorbě nebyly přesně stanoveny. Avšak program je plně funkční a pro demonstraci realizovaného systému bez problémů dostačující.

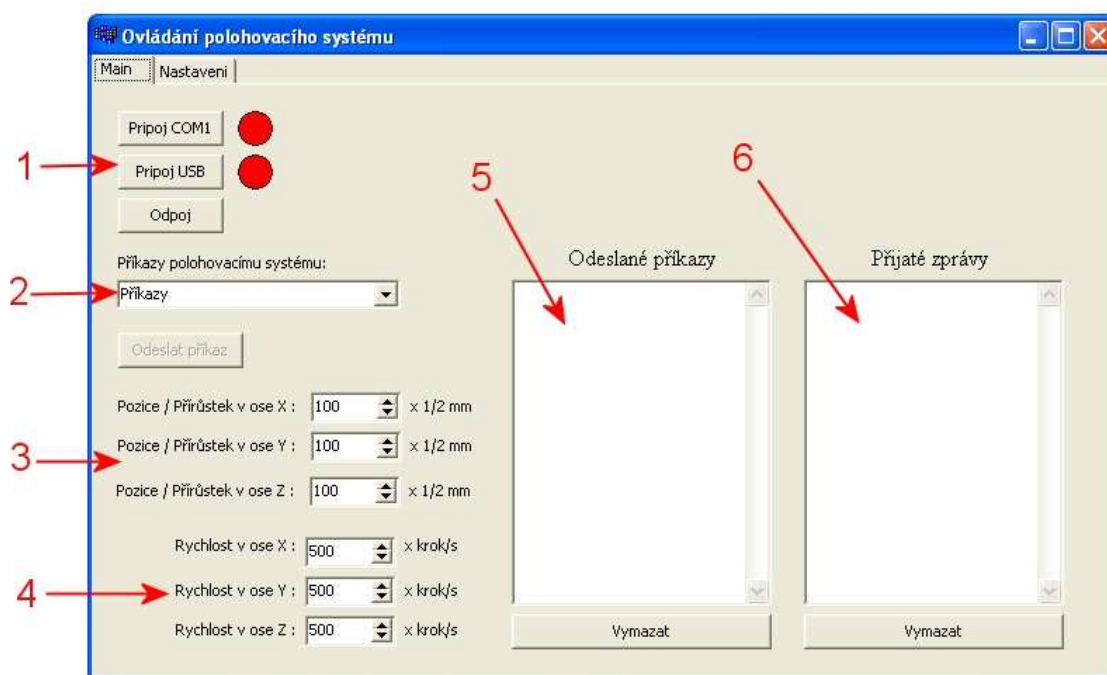
⁹ Virtual Instrument Software Architecture

5.1.2 Grafické rozhraní

Grafické rozhraní aplikace je tvořeno jedním objektem typu WinForm. Neobsahuje mnoho ovládacích prvků, proto není ovládání zařízení pomocí ní nikterak složité.

Ovládací prvky aplikace:

1. Tlačítka pro zahájení/ukončení komunikace s řídicí jednotkou.
2. Nabídka pro výběr příkazu, který má být jednotce předán.
3. Pole pro zadávání požadované velikosti pohybu – buď jako přírůstek polohy nebo absolutní pozice v závislosti na typu příkazu.
4. Pole pro zadávání požadované rychlosti pohybu.
5. Textové pole, do něhož jsou ukládány odeslané příkazy.
6. Textové pole, kam jsou ukládána zpětná hlášení přijatá od jednotky.



Obrázek 5.1 Grafické rozhraní aplikace

Příklad ovládání:

Uživatel fyzicky připojí řídicí jednotku přes rozhraní USB, stiskne tlačítko *Připoj USB*, indikační „dioda“ změní barvu na zelenou, což signalizuje, že aplikace obsadila vybraný port PC a navázala komunikaci s jednotkou.

Necht' je požadavkem uživatele vykonat pohyb v ose X, pomocí relativního zadání polohy. Uživatel tedy vybere z nabídky příkazů (Obrázek 5.1 ukazatel 2) příkaz *POS:X:REL:+/-"pozice":"rychlost"* a do příslušných textových polí (Obrázek 5.1 ukazatel 3) doplní hodnotu požadované změny polohy a velikosti rychlosti. Poté stiskne tlačítko *Odeslat prikaz* a daný příkaz se vypíše (Obrázek 5.1 ukazatel 5). Jednotka ohlásí přijetí příkazu, které je vypsáno v příslušném poli (Obrázek 5.1 ukazatel 6).

V záložce nastavení je možné zvolit mezi dvěma způsoby zadávání parametrů pohybu. První způsob je *Pohyb dle uživatele*, druhým je *Pohyb po přímce*. V prvním případě uživatel zadává mimo parametrů vzdálenosti v jednotlivých osách i rychlost ve všech osách. V případě požadavku pohybu po přímce, uživatel zadává vzdálenosti a pouze jeden parametr rychlosti v kterékoli ose (resp. nadbytečné údaje o rychlosti jsou ignorovány). Řídící jednotka polohovacího systému v tomto případě na základě údaje o rychlosti v jedné ose dopočítá rychlosti v ostatních osách tak, aby se skutečně vykonal pohyb po přímce.

6 Závěr

Účelem této diplomové práce bylo navrhnout a realizovat řízení polohovacího systému, který bude základem automatizovaného měřicího pracoviště pro měření v magnetických polích. Prvým úkolem bylo navrhnout a zkonstruovat řídicí jednotku krokových motorů, které jsou pohonem pro polohování. Tato řídicí jednotka má přijímat příkazy od nadřazeného prvku a ovládat krokový motor. Další úlohou byl návrh a realizace elektronické jednotky, jejímž úkolem je řídit celý polohovací systém. Posledním požadavkem bylo naprogramovat aplikaci pro PC pod Windows, která uživateli zprostředkuje grafické rozhraní pro ovládání celého systému.

Návrhu řídicí jednotky krokových motorů předcházely simulace chování těchto motorů vytvořené v prostředí MATLAB/SIMULINK, jejichž účelem byl výběr nejvhodnějšího způsobu řízení. Realizovaná řídicí jednotka dvoufázových krokových motorů se skládá ze dvou mechanicky oddělených částí. První je logická část, postavená na bázi 8bitového mikrokontroléru ATmega16. Druhá část je výkonová obsahující dva H-můstky z výkonových MOSFET tranzistorů na jejichž diagonálu je připojena příslušná fáze krokového motoru. SW řídicí jednotky ovládá motor pomocí mikrokrokování, umožňuje plynulý rozběh a brzdění s nastavitelnou hodnotou akcelerace a decelerace. V mnoha ohledech (zejména inteligence a možnosti ovládacího SW, nadproudová ochrana, cena atd.) tato jednotka předčí dříve pro tento účel používaný komerční produkt.

Další částí řešení byl návrh řídicí jednotky polohovacího systému. Ke komunikaci jednotky s PC byla zvolena rozhraní RS232 a USB. S ohledem na budoucí rozšíření byl též hardwarově implementován komunikační modul pro bezdrátové rozhraní Bluetooth. V autonomním režimu (bez PC) může uživatel zadávat požadavky pohybu pomocí klávesnice a třířádkového LCD displeje. „Srdcem“ jednotky je mikrokontrolér ATmega128, který přijímá příkazy od uživatele, interpretuje je a po dopočítání potřebných parametrů vysílá přes rozhraní SPI požadavek pohybu podřízeným jednotkám krokových motorů. Mikrokontrolér dále řídí zobrazování na displeji a na obdržený příkaz z PC reaguje zpětným hlášením.

Posledním krokem řešení bylo naprogramovat ovládací aplikaci pro PC. Tato umožňuje uživateli pohodlně vybrat příkaz, který chce se zvolenými parametry řídicí

jednotce zaslat, zajišťuje jeho odeslání a výpis zpětného hlášení. Realizovaný soubor příkazů, jimiž uživatel zadává své požadavky jednotce, vychází ze standardu SCPI.

Ze všech bodů poměrně rozsáhlého zadání nebylo splněno pouze polohování po křivce. Myslím, že tento bod zadání by sám o sobě vydal na samostatnou práci a jeho implementace do SW může být námětem dalšího rozšiřování realizovaného systému.

Ostatní požadavky byly kompletně splněny. Řídící jednotka polohovacího systému i jednotky krokových motorů (3×) jsou odzkoušeny, plně funkční a připraveny pro další použití při konstrukci zmíněného měřicího pracoviště.

Budoucí rozšíření je možné zejména vylepšením ovládací aplikace v PC, která byla napsána pouze za účelem otestování funkce realizovaného celku. Dále je možné doplnit do SW obsluhu Bluetooth komunikačního modulu, čímž by celý systém mohl být ovládán ze vzdáleného zařízení bezdrátově.

Literatura

- [1] Jones, D. W. *Control of Stepping Motors*. [online] The University of Iowa [cit. 20.12.2006]. Dostupné na [www](http://www.cs.uiowa.edu/~jones/step/index.html):
<http://www.cs.uiowa.edu/~jones/step/index.html>

- [2] Kejno, T. *Stepping motors and their microprocessor controls*. 2. vydání, vyd. Belfast: Oxford University Press 1993, 244s. ISBN 0-19-859339-2.

- [3] MICROCON, s.r.o. *Katalogový list krokových motorů*. [online] [cit 20.12.2006]. Dostupné na [www](http://www.microcon.cz/):
<http://www.microcon.cz/>

- [4] ATMEL Corp. Application notes. *AVR151: Setup and use of the SPI*. [online] [cit. 15.2.2007]. Dostupné na [www](http://www.atmel.com/dyn/resources/prod_documents/doc2585.pdf):
http://www.atmel.com/dyn/resources/prod_documents/doc2585.pdf

- [5] JTAG technologies *JTAG Customer references*. [online] [cit 15.2.2007]. Dostupné na [www](http://www.jtag.com/main.php):
<http://www.jtag.com/main.php>

- [6] International Rectifier. *Bootstrap Network Analysis: Focusing on the Integrated Bootstrap Functionality*. [online] [cit. 22.3.2007]. Dostupné na [www](http://www.irf.com/technical-info/appnotes/an-1123.pdf):
<http://www.irf.com/technical-info/appnotes/an-1123.pdf>

- [7] ATMEL Corp. *ATmega16 (Datasheet)* [online] [cit. 12.3.2007]. Dostupné na [www](http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf):
http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf

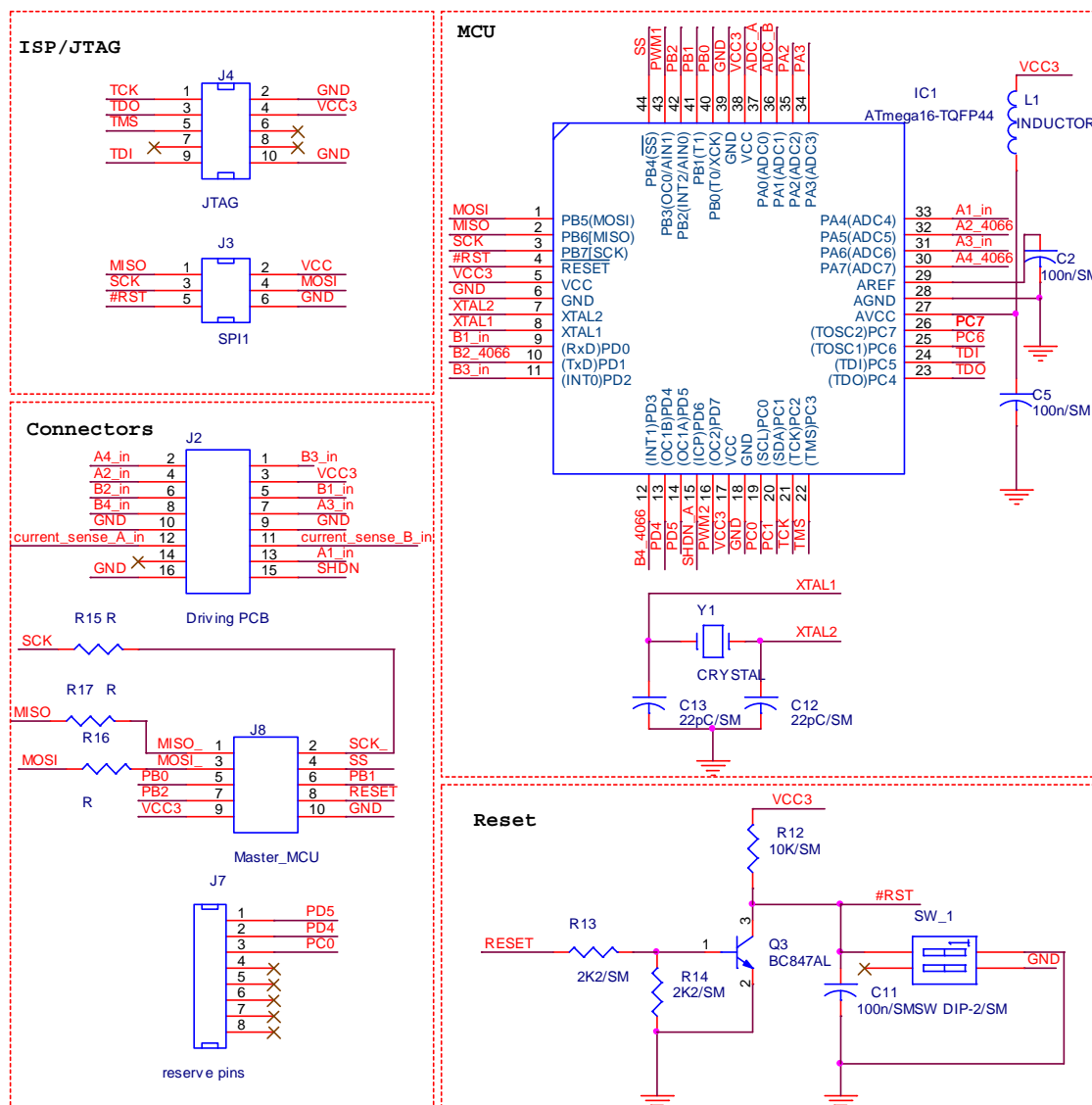
- [8] Austin, D. *Generate stepper-motor speed profiles in real time*. [online] [cit. 12.3.2007]. Dostupné na [www](http://www.embedded.com/showArticle.jhtml?articleID=56800129):
<http://www.embedded.com/showArticle.jhtml?articleID=56800129>

- [9] Haasz, V., Roztočil, J. , Novák, J. *Číslicové měřicí systémy*. 2 vydání. vyd. Praha: Ediční středisko ČVUT 2000, 315 s. ISBN 80-01-02219-6.
- [10] ATMEL Corp. *ATmega128 (Datasheet)* [online] [cit. 20.3.2007].
Dostupné na www:
http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf
- [11] Záhlava, V. *OrCAD 10*. 1.vydání. vyd. Praha: Grada Publishing 2004, 224 s. ISBN 80-247-0904-X.

Příloha A

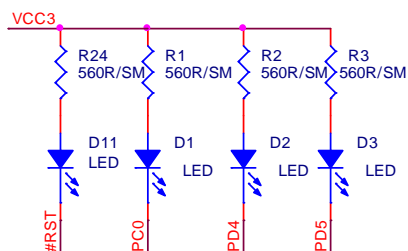
Řídící jednotka krokového motoru – logická část

Schéma, část 1/3

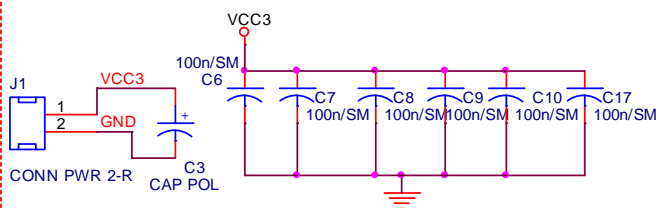


Schéma, část 2/3

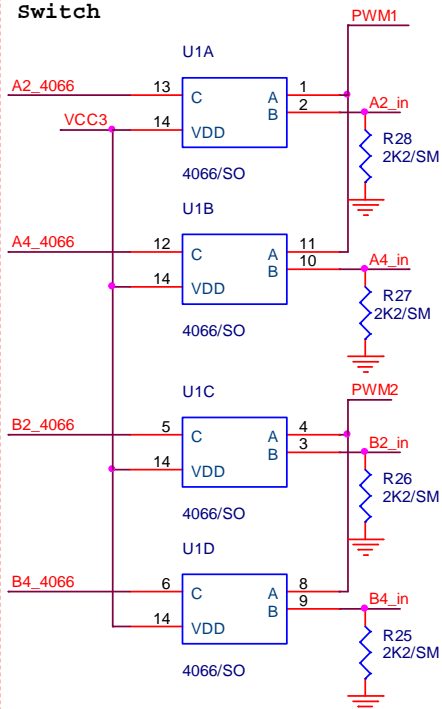
Diodes



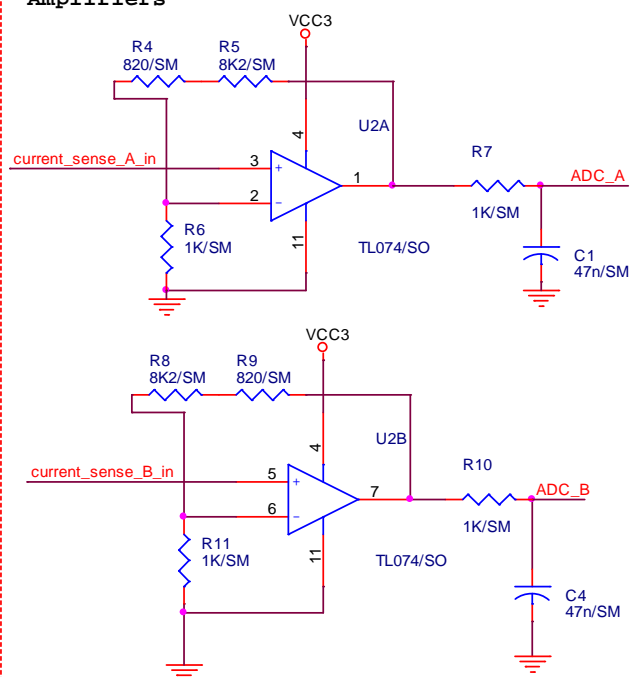
Power and filters



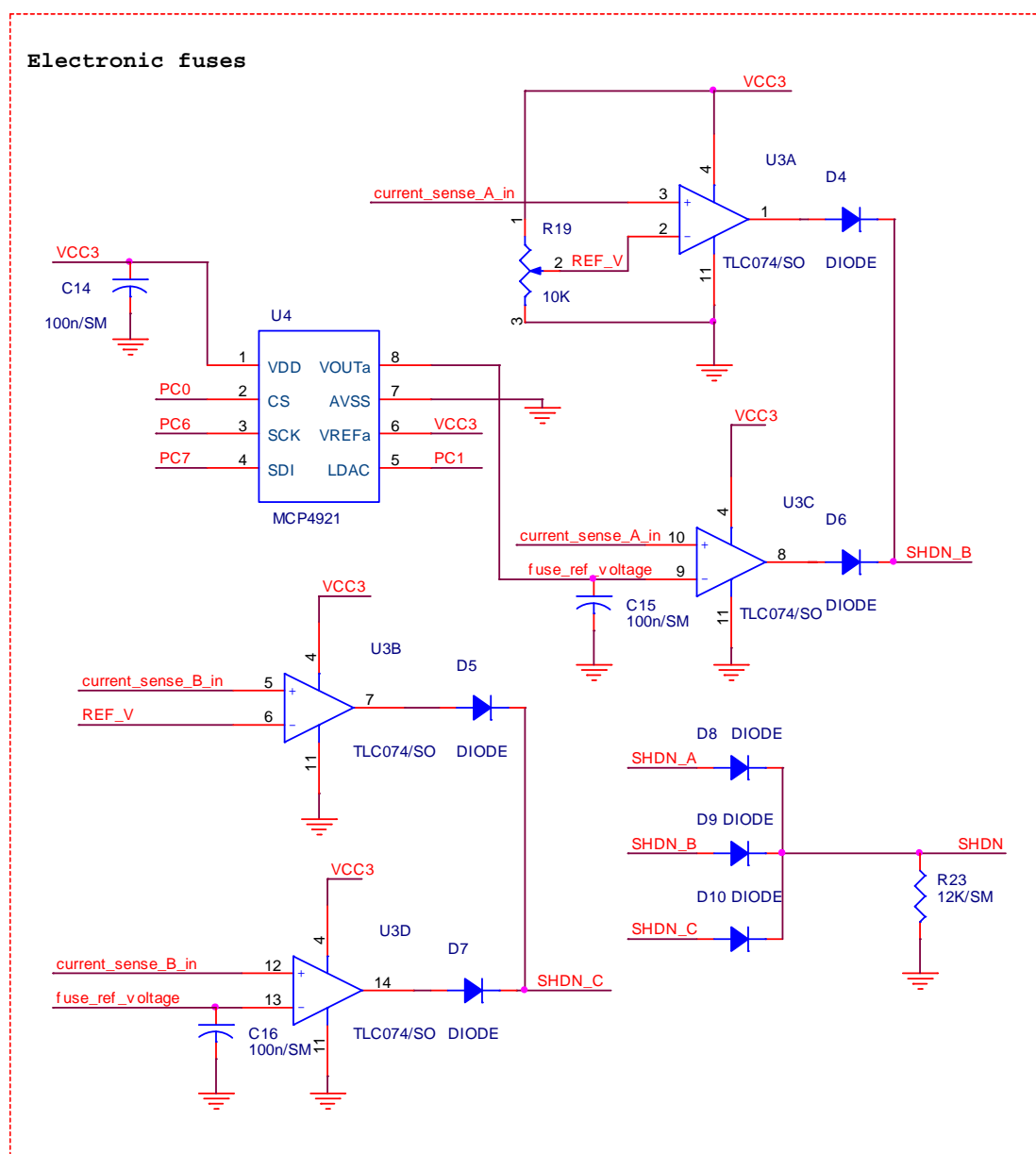
Switch



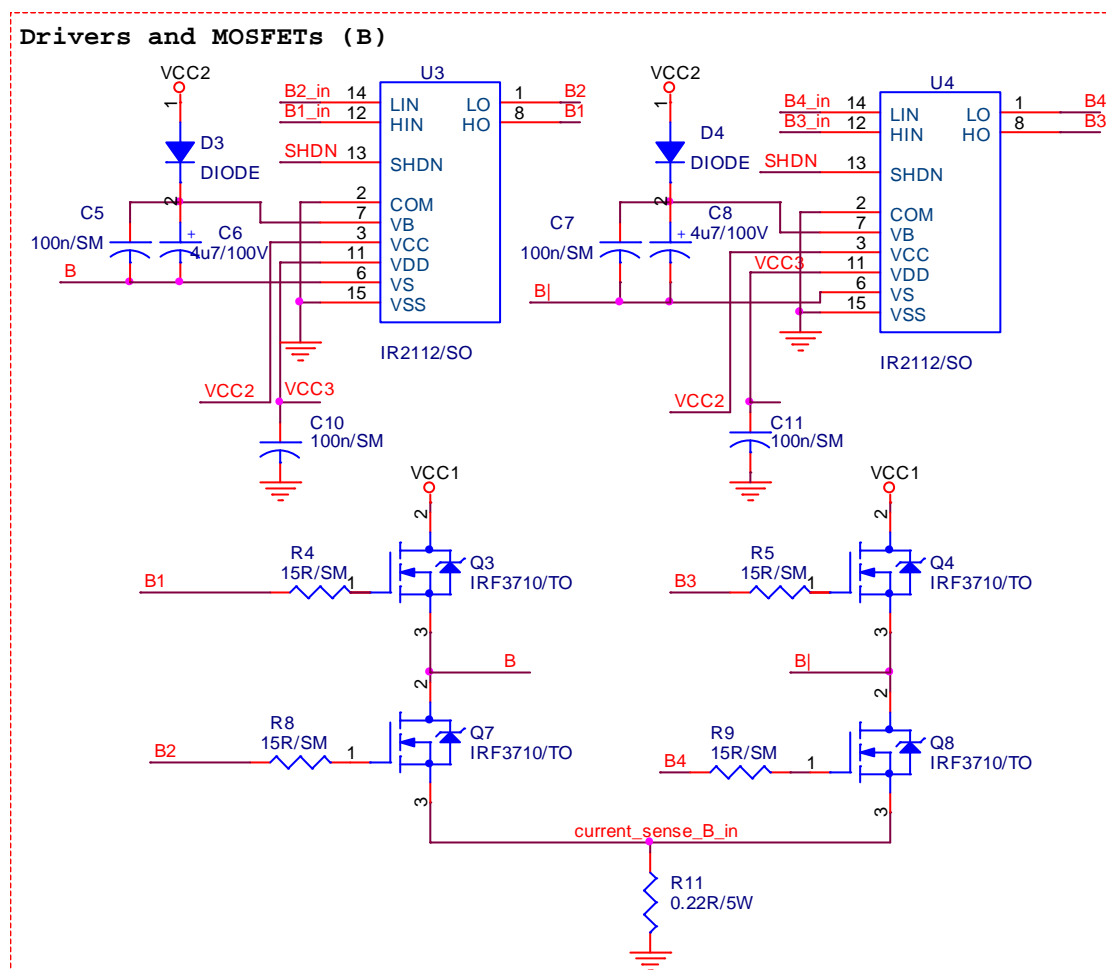
Amplifiers



Schéma, část 3/3



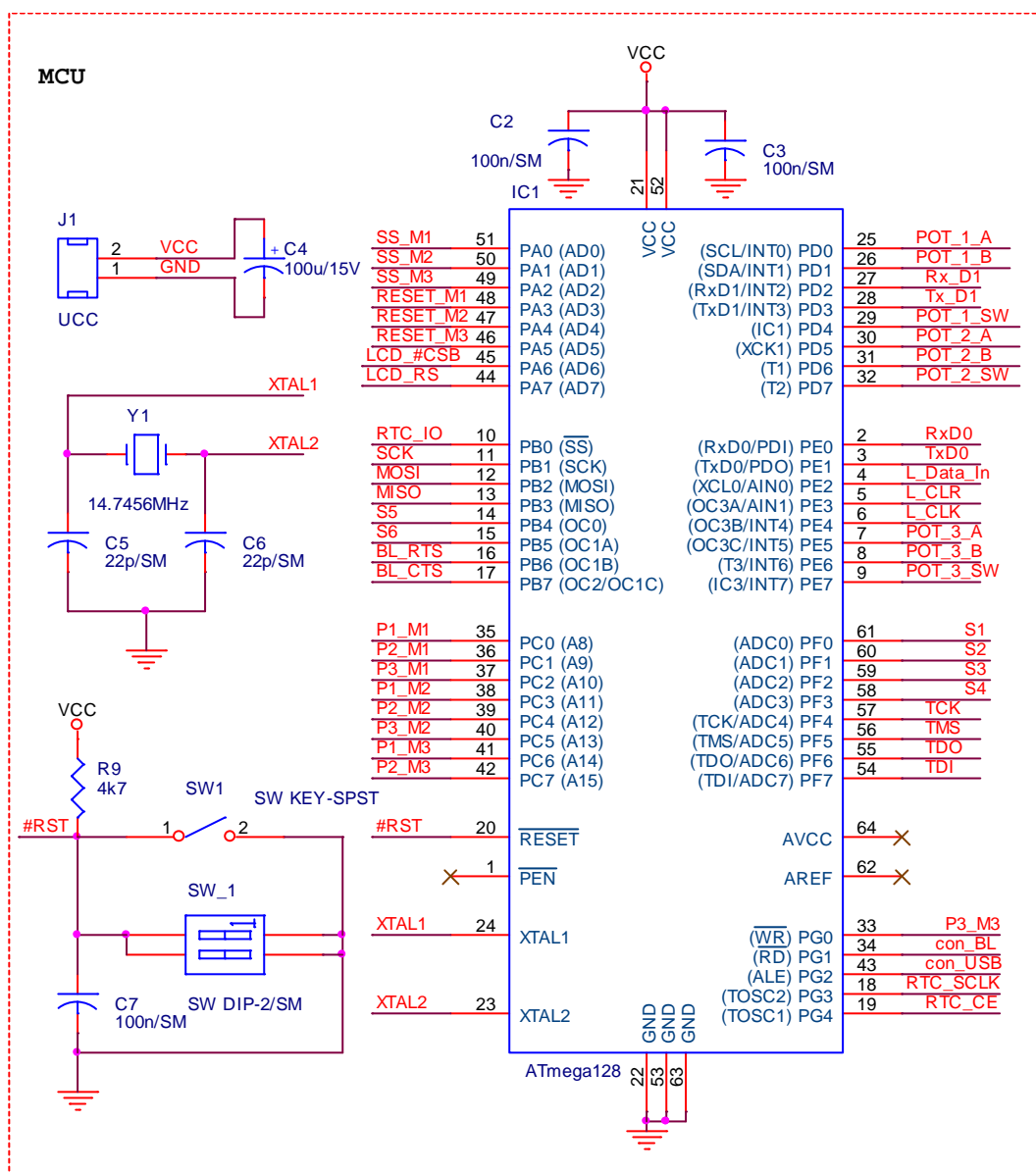
Schéma, část 2/2



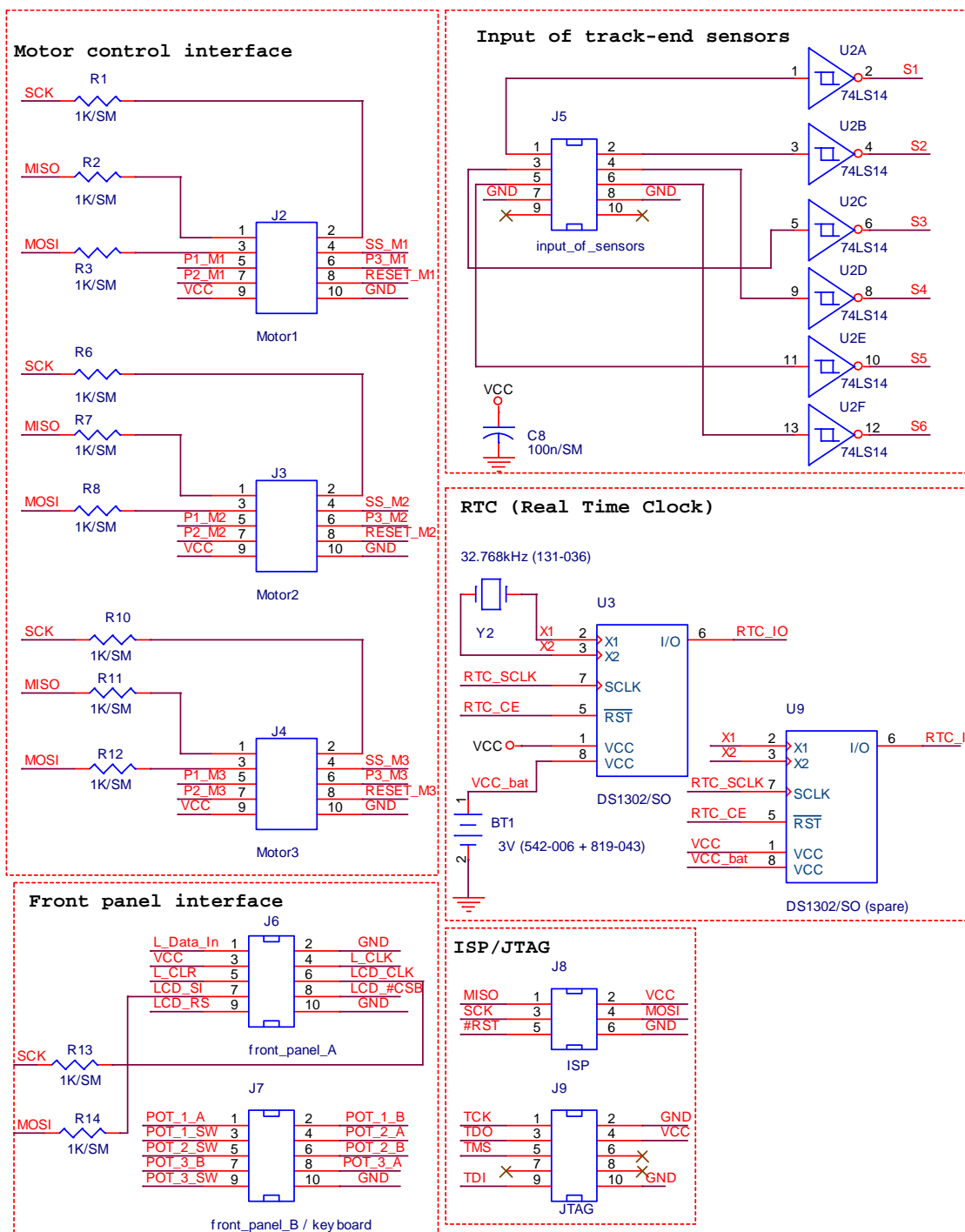
Příloha B

Řídící jednotka polohovacího systému

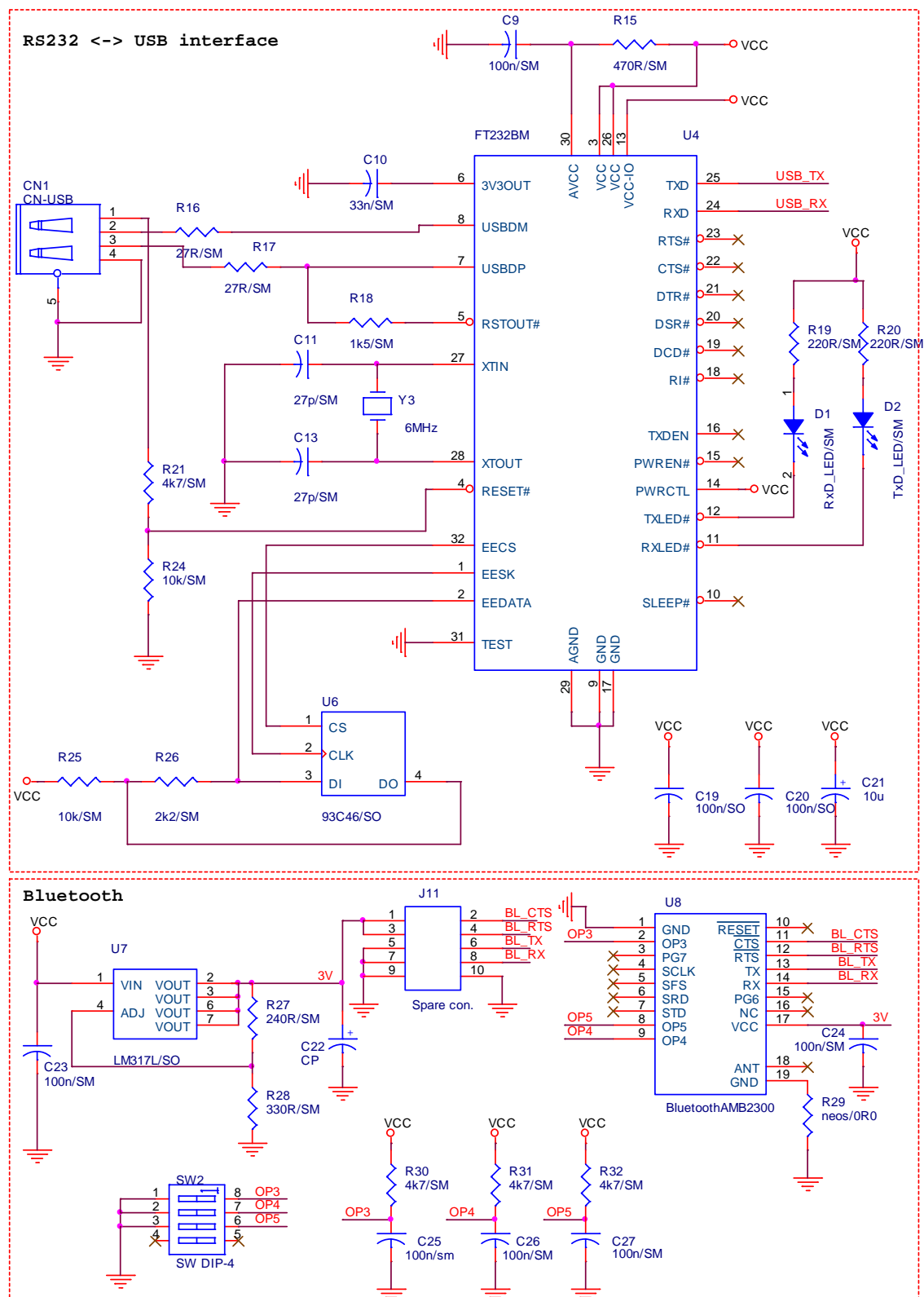
Schéma, část 1/4



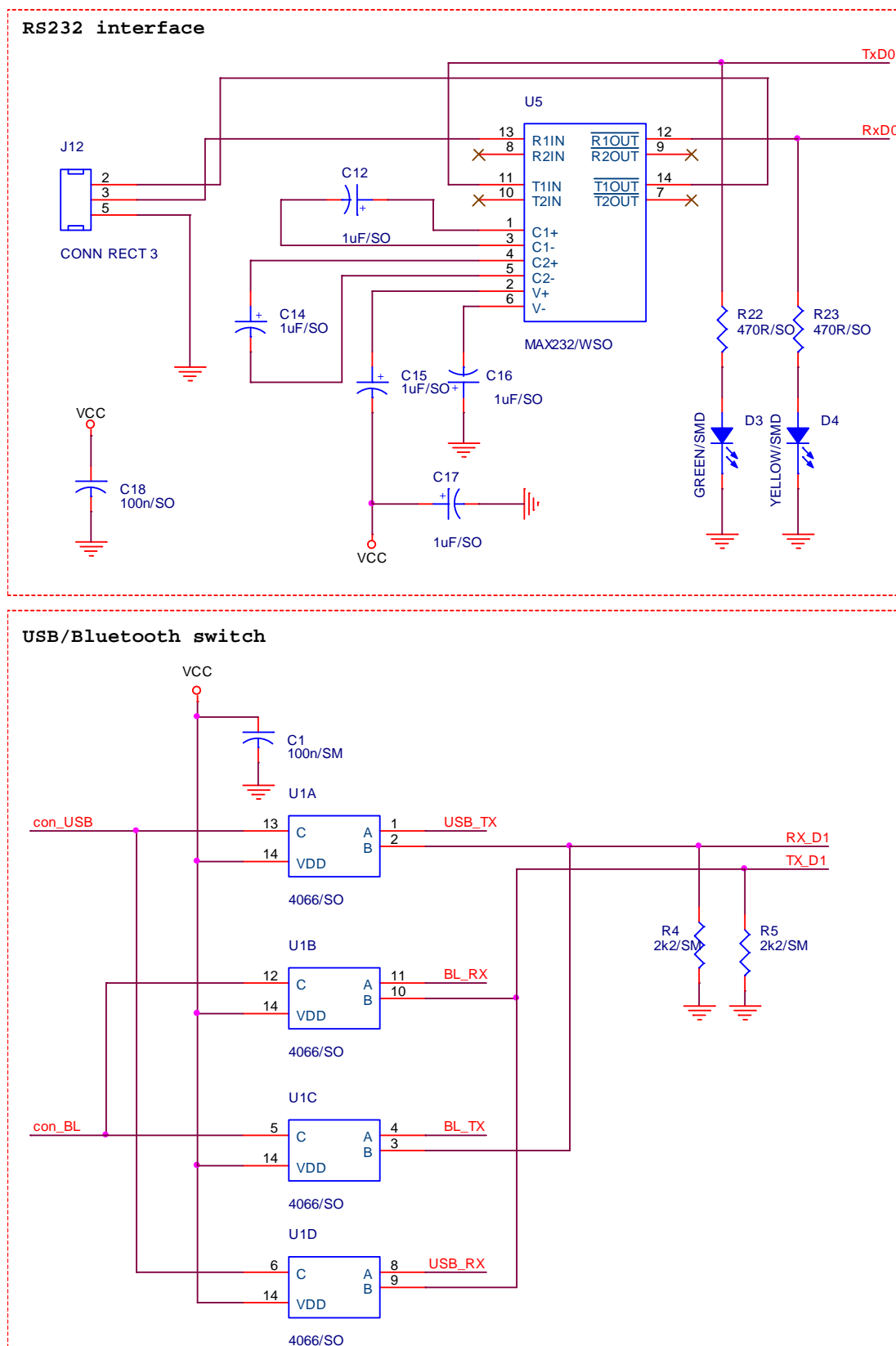
Schéma, část 2/4



Schéma, část 3/4

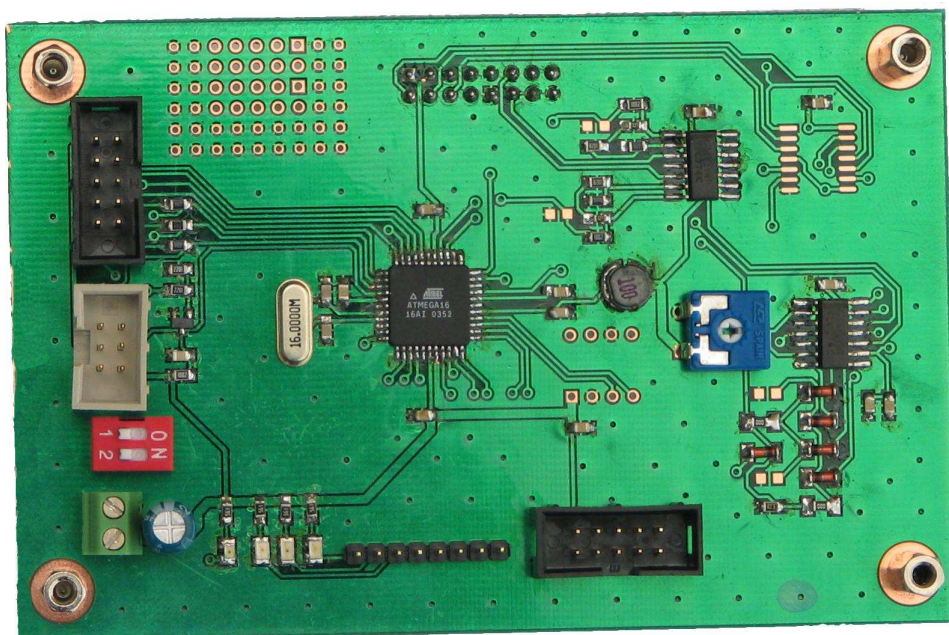


Schéma, část 4/4

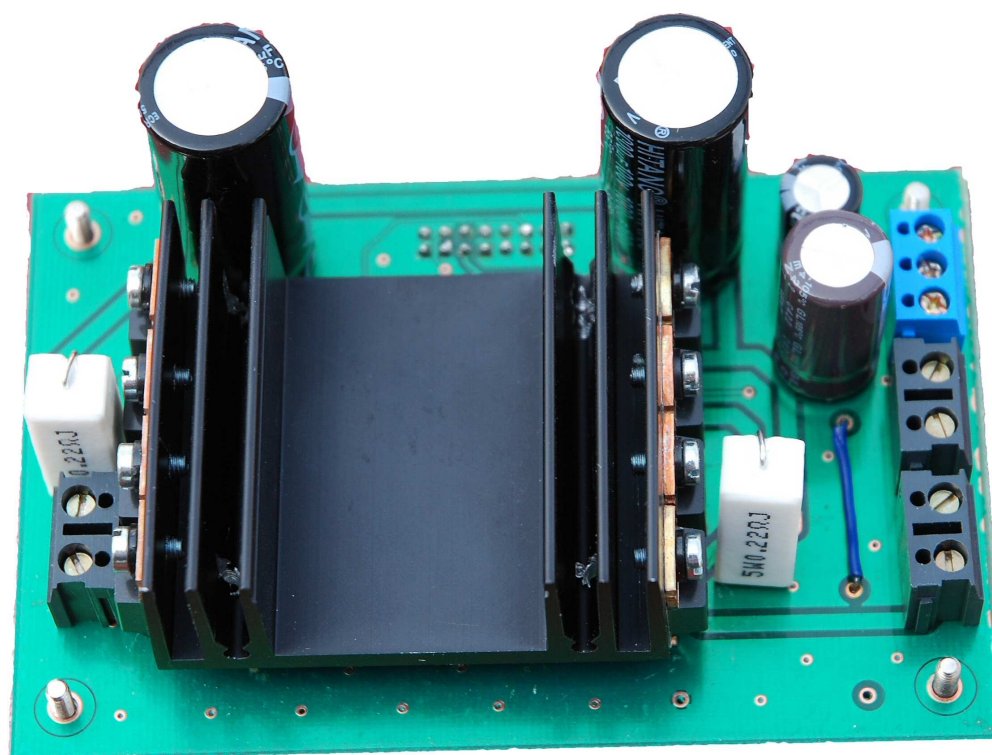


Příloha C

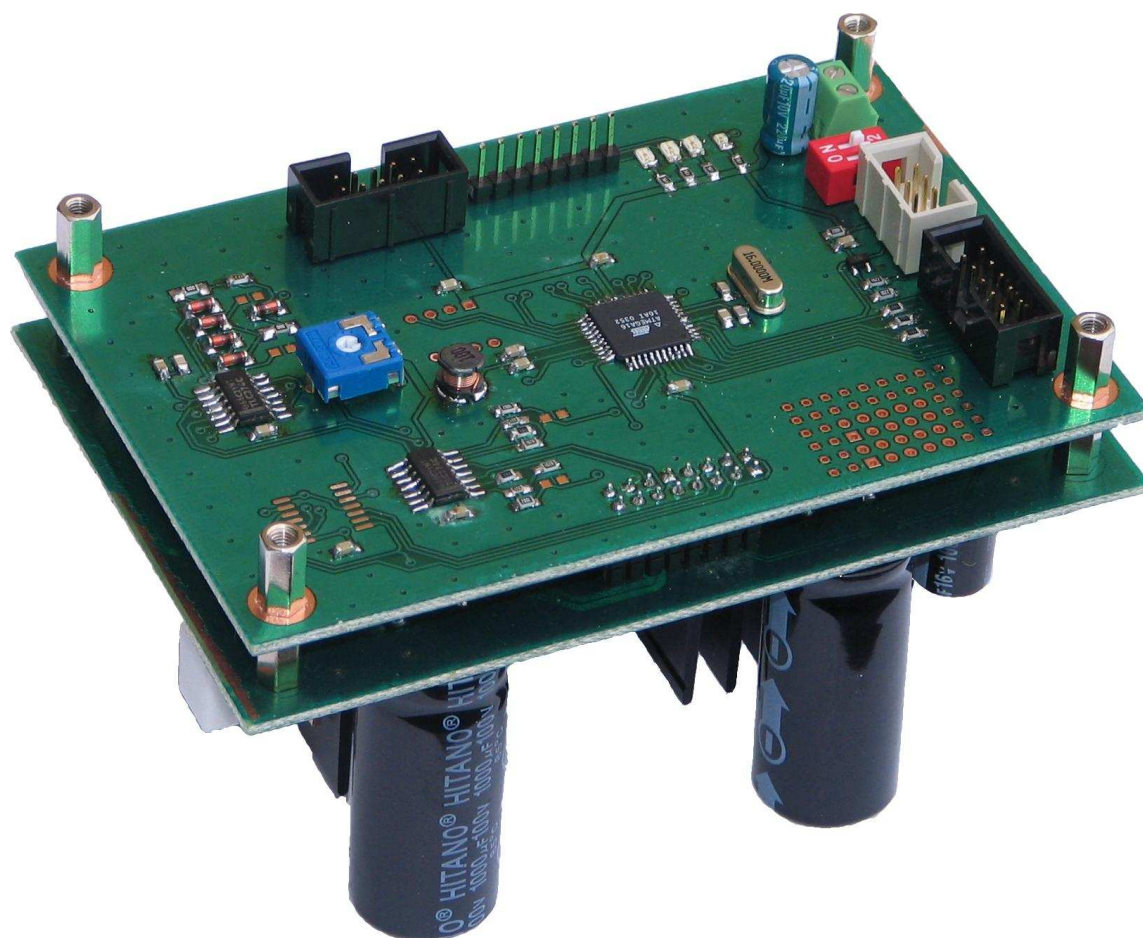
Fotodokumentace



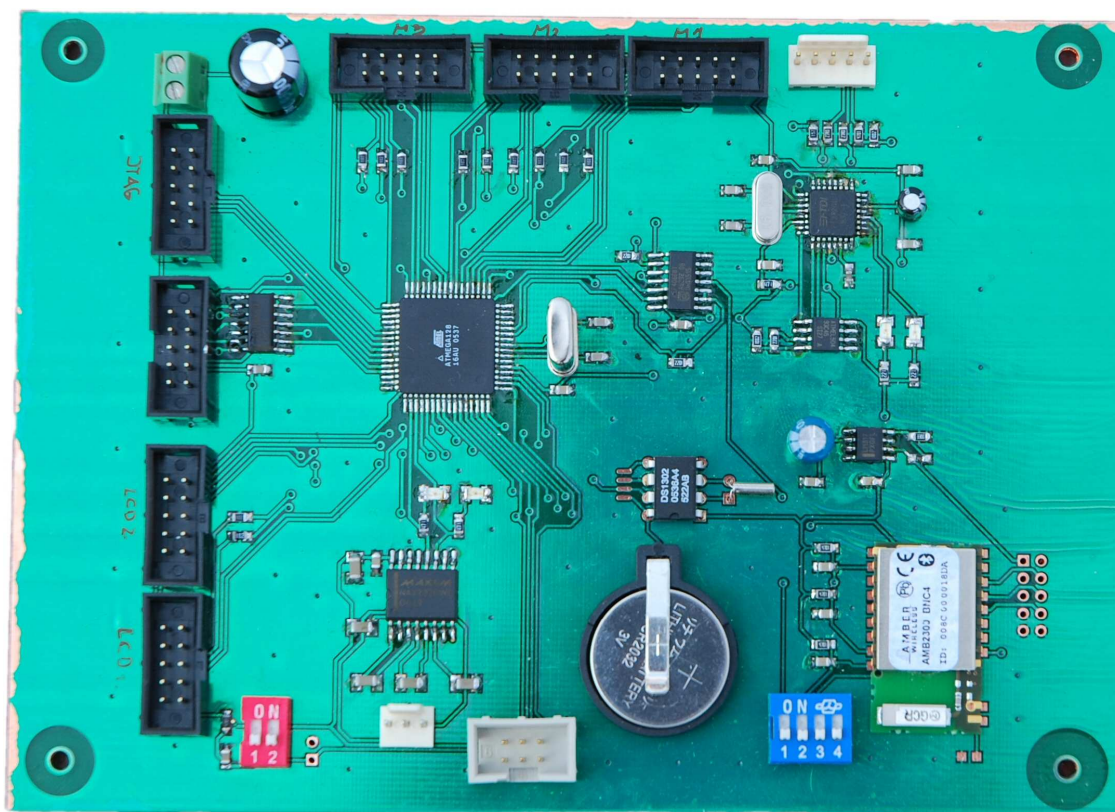
Logická část řídicí jednotky krokového motoru



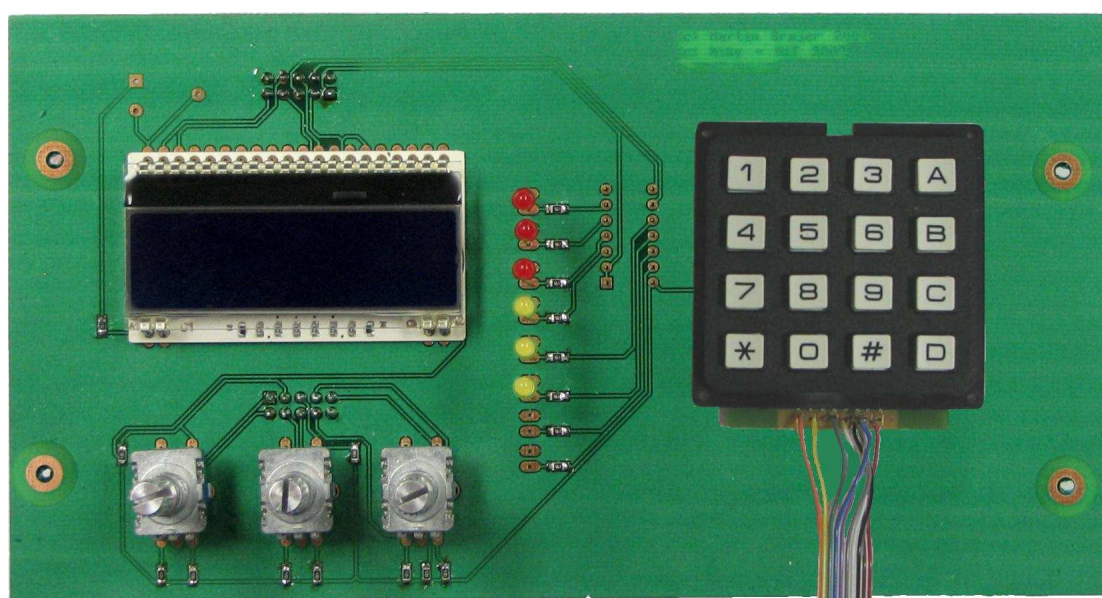
Výkonová část řídicí jednotky krokového motoru



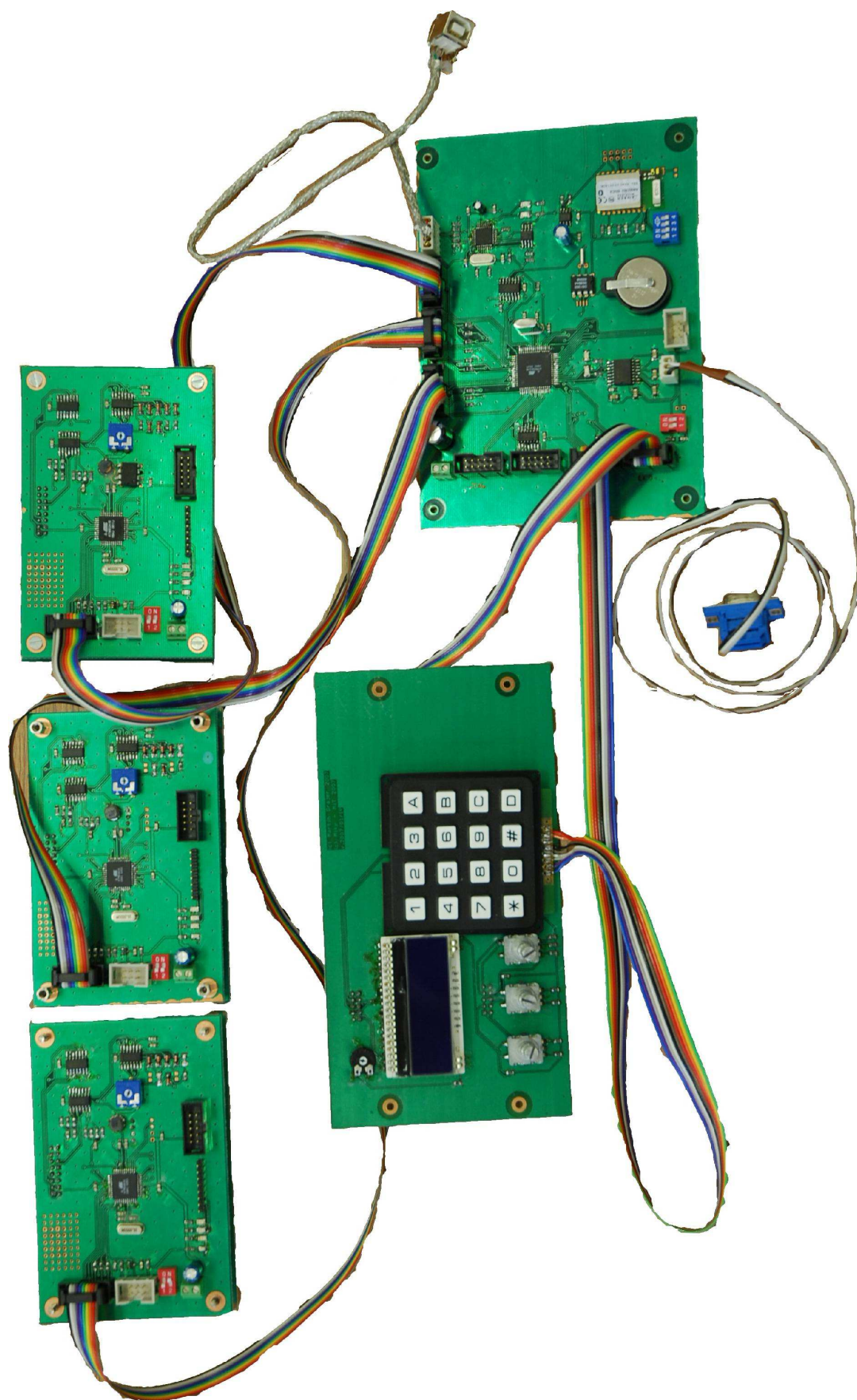
Kompletní řídicí jednotka krokového motoru



Řídící jednotka polohovacího systému



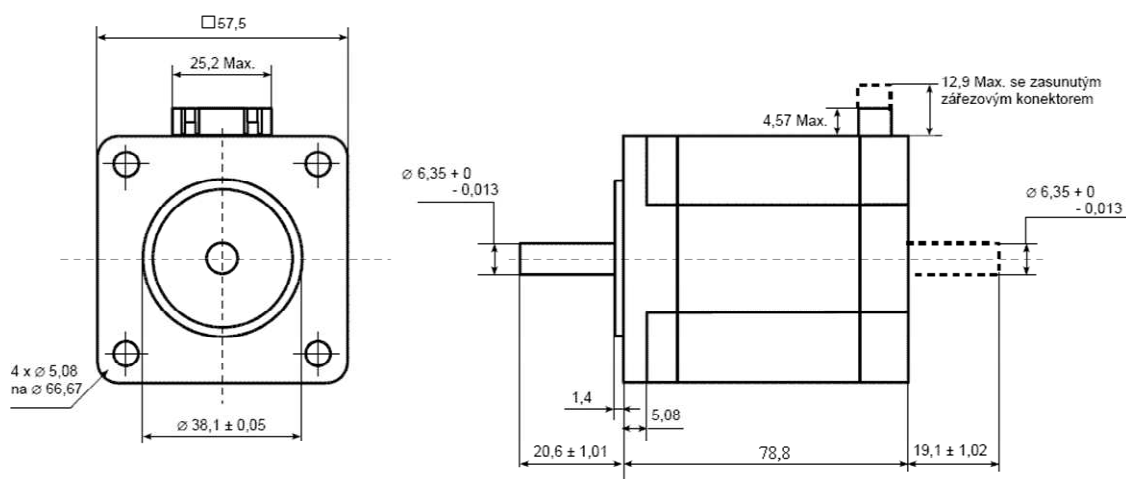
Čelní panel



Kompletní realizovaný HW



Použitý krokový motor

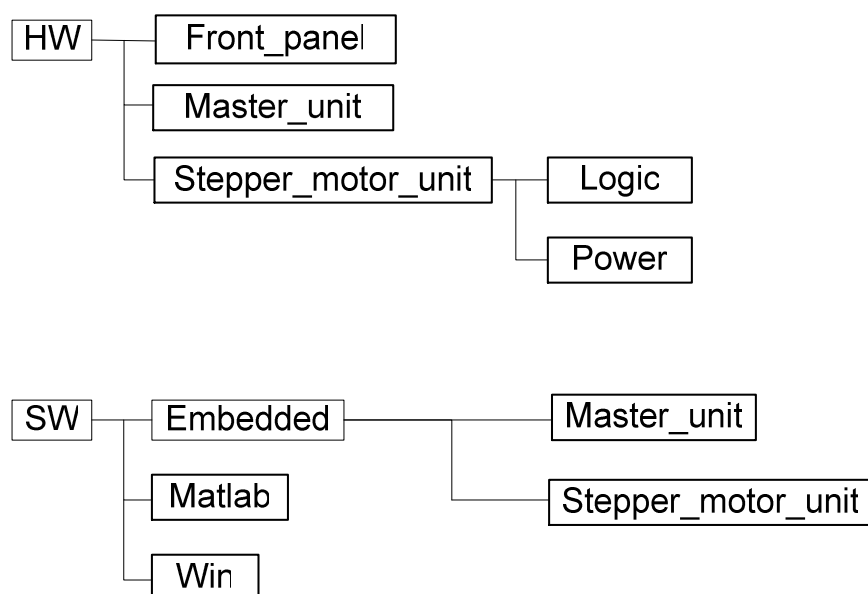


Výkres krokového motoru SM2321-1400 dle [3]

Příloha D

Struktura a obsah přiloženého CD

Přiložené CD obsahuje všechny důležité součásti práce. Složky v oddíle HW obsahují schémata a návrhy desek plošných spojů elektronických částí práce. Složky v oddíle SW obsahují veškeré programy a jejich zdrojové kódy, jenž byly jako součást práce napsány. Ve formátu .pdf je též přiložen text DP.



Struktura složek přiloženého CD